

PPPPPPPPPPPP	RRRRRRRRRRRR	TTTTTTTTTTTTT	SSSSSSSSSSSS	MMM	MMM	BBBBBBBBBBBB			
PPPPPPPPPPPP	RRRRRRRRRRRR	TTTTTTTTTTTTT	SSSSSSSSSSSS	MMM	MMM	BBBBBBBBBBBB			
PPPPPPPPPPPP	RRRRRRRRRRRR	TTTTTTTTTTTTT	SSSSSSSSSSSS	MMM	MMM	BBBBBBBBBBBB			
PPP	PPP	RRR	RRR	TTT	SSS	MMMMMM	MMMMMM	BBB	BBB
PPP	PPP	RRR	RRR	TTT	SSS	MMMMMM	MMMMMM	BBB	BBB
PPP	PPP	RRR	RRR	TTT	SSS	MMMMMM	MMMMMM	BBB	BBB
PPP	PPP	RRR	RRR	TTT	SSS	MMM	MMM	BBB	BBB
PPP	PPP	RRR	RRR	TTT	SSS	MMM	MMM	BBB	BBB
PPP	PPP	RRR	RRR	TTT	SSS	MMM	MMM	BBB	BBB
PPP	PPP	RRR	RRR	TTT	SSS	MMM	MMM	BBB	BBB
PPPPPPPPPPPP	RRRRRRRRRRRR	TTT	SSSSSSSSSS	MMM	MMM	BBBBB6BBBBBBB			
PPPPPPPPPPPP	RRRRRRRRRRRR	TTT	SSSSSSSSSS	MMM	MMM	BBBBBBBBBBBBBB			
PPPPPPPPPPPP	RRRRRRRRRRRR	TTT	SSSSSSSSSS	MMM	MMM	BBBBBBBBBBBBBB			
PPP	RRR	RRR	TTT	SSS	MMM	MMM	BBB	BBB	
PPP	RRR	RRR	TTT	SSS	MMM	MMM	BBB	BBB	
PPP	RRR	RRR	TTT	SSS	MMM	MMM	BBB	BBB	
PPP	RRR	RRR	TTT	SSS	MMM	MMM	BBB	BBB	
PPP	RRR	RRR	TTT	SSS	MMM	MMM	BBB	BBB	
PPP	RRR	RRR	TTT	SSS	MMM	MMM	BBB	BBB	
PPP	RRR	RRR	TTT	SSS	MMM	MMM	BBB	BBB	
PPP	RRR	RRR	TTT	SSS	MMM	MMM	BBB	BBB	
PPP	RRR	RRR	TTT	SSSSSSSSSSSS	MMM	MMM	BBBBBBBBBBBBBB		
PPP	RRR	RRR	TTT	SSSSSSSSSSSS	MMM	MMM	BBBBBBBBBBBBBB		
PPP	RRR	RRR	TTT	SSSSSSSSSSSS	MMM	MMM	BBBBBBBBBBBBBB		

FILEID**SEPARATE

D 13

SEP
V04

SSSSSSSS	EEEEEEEEE	PPPPPPPP	AAAAAA	RRRRRRR	AAAAAA	TTTTTTTTT	EEEEEEEEE
SSSSSSSS	EE	PP PP	AA AA	RR RR	AA AA	TT	EE
SS	EE	PP PP	AA AA	RR RR	AA AA	TT	EE
SS	EE	PP PP	AA AA	RR RR	AA AA	TT	EE
SS	EE	PP PP	AA AA	RR RR	AA AA	TT	EE
SSSSSS	EEEEEEE	PPPPPPPP	AA AA	RRRRRRR	AA AA	TT	EEEEEEE
SSSSSS	EEEEEEE	PPPPPPPP	AA AA	RRRRRRR	AA AA	TT	EEEEEEE
SS	EE	PP	AAAAAAAAA	RR RR	AAAAAAAAA	TT	EE
SS	EE	PP	AAAAAAAAA	RR RR	AAAAAAAAA	TT	EE
SS	EE	PP	AA AA	RR RR	AA AA	TT	EE
SS	EE	PP	AA AA	RR RR	AA AA	TT	EE
SSSSSSSS	EEEEEEEEE	PP	AA AA	RR RR	RR AA	TT	EEEEEEEEE
SSSSSSSS	EEEEEEEEE	PP	AA AA	RR RR	RR AA	TT	EEEEEEEEE

LL	IIIIII	SSSSSSSS
LL	IIIIII	SSSSSSSS
LL	II	SS
LLLLLLLLL	IIIIII	SSSSSSSS
LLLLLLLLL	IIIIII	SSSSSSSS

: F

```
: 1 0001 0 MODULE SEPARATE ( XTITLE,'Print Symbiont -- separation routines'
: 2 0 IDENT = 'VO4-001'
: 3 0 ADDRESSING_MODE (EXTERNAL = GENERAL)
: 4 0 ) =
: 5 1 BEGIN
: 6 1
: 7 1
: 8 1 ****
: 9 1 *
:10 1 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
:11 1 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
:12 1 * ALL RIGHTS RESERVED.
:13 1 *
:14 1 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
:15 1 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
:16 1 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
:17 1 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
:18 1 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
:19 1 * TRANSFERRED.
:20 1 *
:21 1 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
:22 1 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
:23 1 * CORPORATION.
:24 1 *
:25 1 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
:26 1 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
:27 1 *
:28 1 *
:29 1 ****
:30 1 !
:31 1 !
:32 1 ++
:33 1 FACILITY:
:34 1 Print Symbiont.
:35 1
:36 1 ABSTRACT:
:37 1 File and module input routines.
:38 1
:39 1 ENVIRONMENT:
:40 1 VAX/VMS user mode.
:41 1 --
:42 1
:43 1 AUTHOR: Rowland R. Bradley
:44 1
:45 1 CREATION DATE: April 1, 1984
:46 1
:47 1 MODIFIED BY:
:48 1
:49 1 40-001 RRB0006 Rowland R. Bradley 14-Aug-1984
:50 1 Add a <CR> to the page header. This fixes QAR 0682
:51 1 QAR 1737, loss of first line of data of every page.
:52 1
:53 1 3B-005 RRB0005 Rowland R. Bradley 02-Aug-1984
:54 1 Remove the phrase "in this file" in (file) record description.
:55 1 Remove extra spaces in job description after start date
:56 1 and after queued to date. Slightly alter filename algorithm to
:57 1 avoid placing long filename on a single line and NOT displaying
```

SEPARATE
V04-001

Print Symbiont -- separation routines

F 13
16-Sep-1984 02:23:03 VAX-11 Bliss-32 V4.0-742
14-Sep-1984 22:32:26 [PRTSMB.SRC]SEPARATE.B32;2

Page 2
(1)

SEP
V04

: 58 0058 1 !
59 0059 1 !
60 0060 1 !
61 0061 1 !
62 0062 1 !
63 0063 1 !
64 0064 1 !
65 0065 1 !
66 0066 1 !
67 0067 1 !
68 0068 1 !
69 0069 1 !
70 0070 1 !
71 0071 1 !
72 0072 1 !
73 0073 1 !
74 0074 1 !
75 0075 1 !
76 0076 1 !
77 0077 1 !
78 0078 1 !
79 0079 1 !
80 0080 1 !
81 0081 1 !
82 0082 1 !
83 0083 1 !
84 0084 1 !
85 0085 1 !
86 0086 1 !
87 0087 1 !
88 0088 1 !
89 0089 1 !**

the type and version (or vice versa). Add a call to PSMSREAD ITEM DX in GET_QUALIFIERS to test correctness(this is my call and will remain as a test). Modify the file trailer page to avoid an ugly truncation of the input file. Add the acronym 'UIC' to the job description sentence. Add /FEED and /NOFEED to the list of qualifiers in GET_QUALIFIERS.

3B-004 RRB0004 Rowland R. Bradley 04-May-1984
Avoid truncation of the Job Flag Page when burst pages are not specified.

3B-003 GRR0003 Gregory R. Robert 29-Apr-1984
Removed reference to JBC\$_JOBQUEUE until job controller message definitions appear in LIB.L32. Changed ABORTED and REQUEUED words in job sentence to uppercase for emphasis. Considered left and right margins in computing page header size. Create page header only once per task. Fix page header to have file revision date instead of revision number.

3B-002 RRB0002 Rowland R. Bradley 27-Apr-1984
Add dynamic Page_Header routine, display all queue qualifiers in a single phrase, fix form feed and line one problems, print /setup_file & /setup_form, insert job number in burst chars, FIX: footer bar, devicename, receipt box, file desc sentence, trailer header bar, digital logo dynamics, /width, psm\$announce display size, fab valid bit, page setup quals.

3B-001 RRB0001 Rowland R. Bradley 01-Apr-1984
Original version

SEPARATE
V04-001

Print Symbiont -- separation routines

G 13
16-Sep-1984 02:23:03
14-Sep-1984 22:32:26
VAX-11 Bliss-32 V4.0-742
[PRTSMB.SRC]SEPARATE.B32;2

Page 3
(2)

```
; 91      0090 1 LIBRARY 'SY$LIBRARY:LIB';
; 92      0091 1 REQUIRE 'LIBS:SMBDEF';
; 93      0583 1 REQUIRE 'SRC$:SMBREQ';
; 94      1040 1
; 95      1041 1 EXTERNAL ROUTINE
; 96          PSMSBANNER,
; 97          PSMS$READ_ITEM_DX           ! strictly to test behavior
; 98          :
; 99      1045 1
;100     1046 1 LITERAL
;101     1047 1     UPCase_MASK= %B '0100000'.           ! lowercase to uppercase
;102     1048 1     LEAD_MASK = %B '001000'.           ! delete leading blanks
;103     1049 1     SIZE    = 0.
;104     1050 1     ADDR    = 1
;105     1051 1     :
;106     1052 1
;107     1053 1 OWN
;108     1054 1     BURST
;109     1055 1     :
```

SEI
VO

111	1056	1 FORWARD ROUTINE	
112	1057	1 PSMSFILE_BURST	
113	1058	1 PSMSFILE_FLAG	
114	1059	1 PSMSFILE_TRAILER	
115	1060	1 PSMSJOB_BURST	
116	1061	1 PSMSJOB_FLAG	
117	1062	1 PSMSJOB_TRAILER	
118	1063	1 PSMSPAGE_HEADER	
119	1064	1	
120	1065	1 PARSE FILE NAME	
121	1066	1 ALLOCATE PAGE	
122	1067	1 DEALLOCATE PAGE	
123	1068	1 CREATE PAGE HEADER	
124	1069	1 RETURN_FRAME_LENGTH	
125	1070	1 RETURN_FRAME_WIDTH	
126	1071	1 FILL_JOB_FLAG : NOVALUE,	
127	1072	1 FILL_FILE_FLAG : NOVALUE,	
128	1073	1 FILL_JOB_TRAILER : NOVALUE,	
129	1074	1 FILL_FILE_TRAILER : NOVALUE,	
130	1075	1 GET_FORM_SIZE : NOVALUE,	
131	1076	1 GET_REVISION_DATE : NOVALUE,	
132	1077	1 GET_SYSTEM_ANNOUNCEMENT : NOVALUE,	
133	1078	1 GET_JOB_DESCRIPTION : NOVALUE,	
134	1079	1 GET_FILE_DESCRIPTION : NOVALUE,	
135	1080	1 GET_FILE_NAME : NOVALUE,	
136	1081	1 GET_JOB_NAME : NOVALUE,	
137	1082	1 GET_EOJ : NOVALUE,	
138	1083	1 GET_EOF : NOVALUE,	
139	1084	1 GET_ACCOUNTING_INFO : NOVALUE,	
140	1085	1 GET_QUALIFIERS : NOVALUE,	
141	1086	1 GET_QUEUE_QUALIFIERS : NOVALUE,	
142	1087	1 GET_USER_NOTE : NOVALUE,	
143	1088	1 GET_RECEIPT_BOX : NOVALUE,	
144	1089	1 GET_VMS_LOGO : NOVALUE,	
145	1090	1 GET_DIGITAL_LOGO : NOVALUE,	
146	1091	1 GET_RULER_FINE : NOVALUE,	
147	1092	1 GET_RULER_COARSE : NOVALUE,	
148	1093	1 SCROLL_FRAME : NOVALUE,	
149	1094	1 FILL_FRAME : NOVALUE,	
150	1095	1 MOVE_FRAME : NOVALUE,	
151	1096	1 INSERT_FRAME : NOVALUE,	
152	1097	1 CENTER_FRAME : NOVALUE,	
153	1098	1 MERGE_FRAME : NOVALUE,	
154	1099	1 INSERT_NAME_BANNER : NOVALUE,	
155	1100	1 INSERT_FILENAME_BANNER : NOVALUE,	
156	1101	1 INSERT_JOBNUMBER_BANNER : NOVALUE,	
157	1102	1 DELIMIT_STRING : NOVALUE,	
158	1103	1 DELIMIT_STRING_NOT : NOVALUE,	
159	1104	1 FIND_DEST_PTR : NOVALUE,	
160	1105	1 FIND_SOURCE_PTR : NOVALUE,	
161	1106	1 DISCARD : NOVALUE,	
162	1107	1 FILE_OPEN : NOVALUE,	
163	1108	1	
164	1109	1	

scrolls/fills frame w/ string
 fills a frame with chars
 inserts undelimited strings
 inserts delimited strings
 centers undelimited strings
 merges "pure" frames
 inserts banners into frames
 inserts banners into frames
 inserts banners into frames
 delimits strings
 delimits strings
 identifies "pure" frames
 identifies "impure" frames
 discards chars in strings
 determines file accessibility

```
: 166    1110 1 %sbttl 'PSMSFILE_BURST - Print a File Burst Page'  
167    1111 1 Functional Description:  
168    1112 1 This routine controls the creation of the file burst page. The  
169    1113 1 FUNCTION code dictates the action taken in creation.  
170    1114 1  
171    1115 1      OPEN - Allocate and create the File Burst Page  
172    1116 1      READ - Return the current line of the File Burst Page  
173    1117 1      CLOSE - Return the buffer allocated on OPEN  
174    1118 1  
175    1119 1 Formal Parameters:  
176    1120 1      SMB_CONTEXT - Pointer to the SMB  
177    1121 1      USER_CONTEXT - User defined pointer (not used here)  
178    1122 1      FUNCTION - OPEN, READ, CLOSE  
179    1123 1      FUNC_DESC - Pointer to functionally dependent descriptor  
180    1124 1      FUNC_ARG - Pointer to functionally dependent argument  
181    1125 1  
182    1126 1 Implicit Inputs:  
183    1127 1      none  
184    1128 1  
185    1129 1 Implicit Outputs:  
186    1130 1      none  
187    1131 1  
188    1132 1 Returned Value:  
189    1133 1      none  
190    1134 1  
191    1135 1 Side Effects:  
192    1136 1      none  
193    1137 1 --  
194    1138 1 GLOBAL ROUTINE PSMSFILE_BURST ( %SBTTL 'FILE_BURST'  
195    1139 1      SMB_CONTEXT : REF VECTOR,  
196    1140 1      USER_CONTEXT : REF VECTOR,  
197    1141 1      FUNCTION : REF VECTOR,  
198    1142 1      FUNC_DESC : REF VECTOR,  
199    1143 1      FUNC_ARG : REF VECTOR  
200    1144 1      ) =  
201    1145 2 BEGIN  
202  
203    1147 2 LOCAL  
204    1148 2      SCB : REF $BBLOCK,  
205    1149 2      STATUS,  
206    1150 2      FORM_WIDTH,  
207    1151 2      FORM_LENGTH,  
208    1152 2      FORM_SIZE,  
209    1153 2      PAGE_REF : REF PAGE_ARRAY,      ! Declare the pointer to page  
210    1154 2      BUFFER : VECTOR [512,byte],   ! Assume max size 512 bytes  
211    1155 2      STRING_DESC : VECTOR [2];      ! Descriptor to current string  
212    1156 2  
213    1157 2  
214    1158 2      SCB = .SMB_CONTEXT[0];  
215    1159 2  
216    1160 2      ! Check the FUNCTION requested  
217    1161 2  
218    1162 2      SELECTONEU .FUNCTION[0] OF  
219    1163 2      SET  
220    1164 2          [PSMSK_READ]:  
221    1165 2          BEGIN  
222    1166 3          PAGE_REF = .SCB[PSMSA_PAGE_POINTER];
```

```
: 223      1167 3
: 224      1168      ! Output one line at a time
: 225      1169      RETURN PSMS_EOF;
: 226      1170
: 227      1171      FUNC_DESC[SIZE] = .SCB[PSMSL_PAGE_WIDTH];
: 228      1172      FUNC_DESC[ADDR] = PAGE_REF[0,.SCB[PSMSL_RECORD_NUMBER],
: 229      1173          .SCB[PSMSL_PAGE_WIDTH]];
: 230      1174
: 231      1175      ! adjust pointer
: 232      1176      FUNC_DESC[SIZE] = DELIMIT_STRING_NOT (.FUNC_DESC[ADDR],
: 233      1177          %CHAR[32], .FUNC_DESC[SIZE]);
: 234      1178
: 235      1179 2
: 236      1180
: 237      1181 2 [PSMSK_OPEN]:
: 238      1182      BEGIN
: 239      1183
: 240      1184      GET_FORM_SIZE (.SCB);           ! Returns the WidthxLength
: 241      1185
: 242      1186
: 243      1187      FORM_WIDTH     = .SCB[PSMSL_PAGE_WIDTH];
: 244      1188      FORM_LENGTH    = .SCB[PSMSL_PAGE_LENGTH];
: 245      1189
: 246      1190      RETURN_IF_ERROR_(ALLOCATE_PAGE( .SCB)); ! Get the page of memory
: 247      1191
: 248      1192      PAGE_REF = .SCB[PSMSA_PAGE_POINTER];   ! My local page pointer
: 249      1193
: 250      1194      ! Allocate the buffer for "GET_xxx" Routines
: 251      1195
: 252      1196      STRING_DESC[SIZE] = %ALLOCATION(BUFFER);       ! allocate for routines
: 253      1197      STRING_DESC[ADDR] = BUFFER;                  ! init address
: 254      1198
: 255      1199      ! No Form_feed for the burst page
: 256      1200
: 257      1201      ! Format the page identically to File Flag
: 258      1202      ! Standard Burst Page 132x66: text covers rows 2 through 60,
: 259      1203      ! translated to frames... ref starts at 2 and length is 58.
: 260      1204      FILL_FILE_FLAG(
: 261      1205          .SCB,
: 262      1206          PAGE_REF[0,2,.FORM_WIDTH],
: 263      1207          .FORM_WIDTH,
: 264      1208          .FORM_LENGTH - 6 - 2 );           ! bottom margin is 6
: 265      1209          ;                                ! 2 spaces at the top
: 266      1210 2
: 267      1211 2 END;
: 268      1212 2 [PSMSK CLOSE];                      ! Return the Page of Memory
: 269      1213 2      RETURN_IF_ERROR_(DEALLOCATE_PAGE(.SCB));
: 270      1214 2
: 271      1215 2 [OTHERWISE]:
: 272      1216 2      RETURN PSMS_FUNNOTSUP;
: 273      1217 2
: 274      1218 2      TES; ! case .function
: 275      1219 2
: 276      1220 2      SSS_NORMAL
: 277      1221 1 END;
```

SEPARATE
V04-001

Print Symbiont -- separation routines
FILE_BURST

K 13

16-Sep-1984 02:23:03
14-Sep-1984 22:32:26

VAX-11 Bliss-32 V4.0-742
[PRTSMB.SRC] SEPARATE.B32;2

Page 7
(4)

				.TITLE	SEPARATE Print Symbiont -- separation routines
				.IDENT	\V04-001\
				.PSECT	DATA,NOEXE,2
			00000 BURST:	.BLKB	4
				.EXTRN	BAS\$EDIT, LBR\$CLOSE
				.EXTRN	LBR\$GET RECORD, LBR\$INI CONTROL
				.EXTRN	LBR\$LOOKUP KEY, LBR\$OPEN
				.EXTRN	LBR\$RET RMSSTV, LBR\$SET_LOCATE
				.EXTRN	LIB\$TRIM FILESPEC
				.EXTRN	LIB\$GET VM, LIB\$FREE_VM
				.EXTRN	STR\$ANALYZE_SDESC
				.EXTRN	STR\$ANALYZE_SDESC R1
				.EXTRN	STR\$APPEND, STR\$CONCAT
				.EXTRN	STR\$COPY DX, STR\$COPY R
				.EXTRN	STR\$FREET DX, STR\$FREE1_DX_R4
				.EXTRN	STR\$GET1 DX, STR\$LEFT
				.EXTRN	STR\$PREFIX, STR\$RIGHT
				.EXTRN	PSMS\$ HANGUP DISPATCH ENTRY
				.EXTRN	PSMS\$ BUFFEROF, PSMS\$ EOF
				.EXTRN	PSMS\$ ESCAPE, PSMS\$ FLUSH
				.EXTRN	PSMS\$ FUNNOTSUP, PSMS\$ INVITMCOD
				.EXTRN	PSMS\$ INVVMSOSC, PSMS\$ MODNOTFND
				.EXTRN	PSMS\$ NEWPAGE, PSMS\$ NOFILEID
				.EXTRN	PSMS\$ OSCTOOLON, PSMS\$ PENDING
				.EXTRN	PSMS\$ SUSPEND, PSMS\$ TOOMANYLEV
				.EXTRN	SMBS\$ INVSTMNBR, SMBS\$ INVSTRLEV
				.EXTRN	SMBS\$ NOMOREITEMS
				.EXTRN	PSMS\$BANNER, PSMS\$READ_ITEM_DX
				.PSECT	CODE,NOWRT,2
				.ENTRY	PSMS\$FILE_BURST, Save R2,R3,R4,R5
			5E FDF8 003C 00000	MOVAB	-520(SP), SP
			52 04 BC 9E 00002	MOVL	@SMB_CONTEXT, SCB
			50 OC BC D0 00007	MOVL	@FUNCTION, R0
			05 50 D1 0000F	CMPL	R0, #5
			41 12 00012	BNEQ	2\$
			50 01FC C2 D0 00014	MOVL	508(SCB), PAGE_REF
			C2 06 C3 00019	SUBL3	#6, 504(SCB), R0
			50 026C C2 D1 0001F	CMPL	620(SCB), R0
			08 15 00024	BLEQ	1\$
			50 00000000G 8F D0 00026	MOVL	#PSMS\$EOF, R0
			04 0002D	RET	
			53 10 AC D0 0002E	1\$: MOVL	FUNC_DESC, R3
			63 0200 C2 D0 00032	MOVL	512(SCB), (R3)
			50 C2 C5 00037	MULL3	512(SCB), 620(SCB), R0
			55 C1 0003F	ADDL3	PAGE_REF, R0, 4(R3)
			63 DD 00044	PUSHL	(R3)
			20 DD 00046	PUSHL	#32
			04 A3 DD 00048	PUSHL	4(R3)
			03 FB 0004B	CALLS	#3, DELIMIT_STRING_NOT
			50 DO 00050	MOVL	R0, (R3)
			58 11 00053	BRB	5\$
			04 50 D1 00055	CMPL	R0, #4

Print Symbiont -- separation routines				16-Sep-1984 02:23:03	VAX-11 Bliss-32 V4.0-742	Page 8 (4)
FILE_BURST				14-Sep-1984 22:32:26	[PRTSMB.SRC]SEPARATE.B32;2	
0000V	CF		3B	12 00058	BNEQ 3\$	
	53	0200	52	DD 0005A	PUSHL SCB	1184
	54	01FB	01	FB 0005C	CALLS #1, GET_FORM_SIZE	
			C2	DO 00061	MOVL 512(SCBT), FORM_WIDTH	1187
			C2	DO 00066	MOVL 504(SCB), FORM_LENGTH	1188
			52	DD 0006B	PUSHL SCB	1190
0000V	CF		01	FB 0006D	CALLS #1, ALLOCATE_PAGE	
	38	01FC	50	E9 00072	BLBC STATUS, 6\$	1192
	55	0200	C2	DO 00075	MOVL 508(SCB), PAGE_REF	
	6E	08	8F	3C 0007A	MOVZWL #512, STRING_DESC	1196
04	AE	F8	AE	9E 0007F	MOVAB BUFFER, STRING_DESC+4	
			A4	9F 00084	PUSHAB -8(FORM_LENGTH)	1197
			53	DD 00087	PUSHL FORM_WIDTH	1208
			6543	3F 00089	PUSHAW (PAGE_REF)[FORM_WIDTH]	1207
			52	DD 0008C	PUSHL SCB	1206
0000V	CF		04	FB 0008E	CALLS #4, FILL_FILE_FLAG	
			18	11 00093	BRB 5\$	1162
	02		50	D1 00095	3\$: CMPL R0, #2	
			08	12 00098	BNEQ 4\$	1211
0000V	CF		52	DD 0009A	PUSHL SCB	
	09	01	FB 0009C	CALLS #1, DEALLOCATE_PAGE	1212	
		50	E8 000A1	BLBS STATUS, 5\$		
			04	000A4	RET	
	50 00000000G	8F	DO 000A5	4\$: MOVL #PSMS_FUNNOTSUP, R0	1215	
			04	000AC	RET	
	50	01	DO 000AD	5\$: MOVL #1, R0		
			04	000B0	6\$: RET	1221

; Routine Size: 177 bytes, Routine Base: CODE + 0000

279 1222 1 %sbttl 'PSMSFILE_FLAG - Print a File Flag Page'
280 1223 1 Functional Description:
281 1224 1 This routine controls the creation of the file flag page. The
282 1225 1 FUNCTION code dictates the action taken in creation.
283 1226 1 FUNCTION:
284 1227 1 OPEN - Allocate and create the file Flag Page
285 1228 1 READ - Return the current line of the file Flag Page
286 1229 1 CLOSE - Return the buffer allocated on OPEN
287 1230 1
288 1231 1 Formal Parameters:
289 1232 1 SMB_CONTEXT - Pointer to the SMB
290 1233 1 USER_CONTEXT - User defined pointer (not used here)
291 1234 1 FUNCTION - OPEN, READ, CLOSE
292 1235 1 FUNC_DESC - Pointer to functionally dependent descriptor
293 1236 1 FUNC_ARG - Pointer to functionally dependent argument
294 1237 1
295 1238 1 Implicit Inputs:
296 1239 1 none
297 1240 1
298 1241 1 Implicit Outputs:
299 1242 1 none
300 1243 1
301 1244 1 Returned Value:
302 1245 1 none
303 1246 1
304 1247 1 Side Effects:
305 1248 1 none
306 1249 1 --
307 1250 1
308 1251 1 GLOBAL ROUTINE PSMSFILE_FLAG (%SBTTL 'FILE_FLAG'
309 1252 1 SMB_CONTEXT : REF VECTOR,
310 1253 1 USER_CONTEXT : REF VECTOR,
311 1254 1 FUNCTION : REF VECTOR,
312 1255 1 FUNC_DESC : REF VECTOR,
313 1256 1 FUNC_ARG : REF VECTOR
314 1257 1) =
315 1258 2 BEGIN
316 1259 2
317 1260 2 LOCAL
318 1261 2 SCB : REF SBBLOCK,
319 1262 2 STATUS,
320 1263 2 FORM_WIDTH,
321 1264 2 FORM_LENGTH,
322 1265 2 FORM_SIZE,
323 1266 2 BUFFER : VECTOR [512,byte], | Assume max size 512 bytes
324 1267 2 STRING_DESC : VECTOR [2], | Descriptor to current string
325 1268 2 PAGE_REF : REF PAGE_ARRAY; | Declare the pointer
326 1269 2 | to page
327 1270 2
328 1271 2 SCB = .SMB_CONTEXT[0];
329 1272 2 ! Check the FUNCTION requested
330 1273 2
331 1274 2 SELECTONEU .FUNCTION[0] OF
332 1275 2 SET
333 1276 2 [PSMSK READ]:
334 1277 2 BEGIN
335 1278 3

336 1279 3 PAGE_REF = .SCB[PSMSA_PAGE_POINTER];
337 1280 3 ! Output one line at a time
338 1281 4 IF (.SCB[PSMSL_RECORD_NUMBER] GTR .SCB[PSMSL_PAGE_LENGTH])
339 1282 3 OR
340 1283 4 (NOT .SEPARATE_FLAG_(FILE_BURST) AND
341 1284 4 .SCB[PSMSL_RECORD_NUMBER] GTR (.SCB[PSMSL_PAGE_LENGTH] - 6))
342 1285 3 THEN
343 1286 3 RETURN PMS_EOF;
344 1287 3
345 1288 3
346 1289 3 FUNC_DESC[SIZE] = .SCB[PSMSL_PAGE_WIDTH];
347 1290 3 FUNC_DESC[ADDR] = PAGE_REF[0..SCB[PSMSL_RECORD_NUMBER],
348 1291 3 .SCB[PSMSL_PAGE_WIDTH]];
349 1292 3 ! adjust pointer
350 1293 3 FUNC_DESC[SIZE] = DELIMIT_STRING_NOT (.FUNC_DESC[ADDR],
351 1294 3 XCHART32), .FUNC_DESC[SIZE]);
352 1295 3
353 1296 2 END;
354 1297 2
355 1298 2 [PSMSK OPEN]:
356 1299 2 BEGIN
357 1300 2
358 1301 2 GET_FORM_SIZE (.SCB); ! Returns the WidthxLength
359 1302 2
360 1303 2
361 1304 2 FORM_WIDTH = .SCB[PSMSL_PAGE_WIDTH];
362 1305 2 FORM_LENGTH = .SCB[PSMSL_PAGE_LENGTH];
363 1306 2
364 1307 2 RETURN_IF_ERROR_(ALLOCATE_PAGE(.SCB)); ! Get the page of memory
365 1308 2
366 1309 2 PAGE_REF = .SCB[PSMSA_PAGE_POINTER]; ! My local page pointer
367 1310 2
368 1311 2 ! Always start at top of page
369 1312 2
370 1313 2 PAGE_REF[0,0..FORM_WIDTH] = PMSK_CHAR_FF; ! form feed in 0 pos.
371 1314 2
372 1315 2 ! Standard Flag Page 132x66: text covers rows 1 through 58,
373 1316 2 translated to frames... ref starts at 1 and length is 57.
374 1317 2 FILL_FILE_FLAG(.SCB,
375 1318 2 PAGE_REF[0,1..FORM_WIDTH],
376 1319 2 .FORM_WIDTH,
377 1320 2 .FORM_LENGTH - 6 - 2 - 1); ! total form length...
378 1321 2 ! 6 burst, 2 spaces
379 1322 2 ! top margin of 1
380 1323 2 ! Burst always appears on the Flag page, starting at page length - 5
381 1324 2 and continuing to page length. This leaves a two blank lines
382 1325 2 between file flag footer bar and file burst header bar.
383 1326 2 This IS the right way to perform a BURST over the crease !
384 1327 2
385 1328 3 IF (.SEPARATE_FLAG_(FILE_BURST)) THEN
386 1329 4 BEGIN
387 1330 4 STRING_DESC[SIZE] = XALLOCATION(BUFFER);
388 1331 4 STRING_DESC[ADDR] = BUFFER;
389 1332 4
390 1333 4 ! determine the correct size of the string to insert
391 1334 4
392 1335 4 GET_VMS_LOGO

```

393    1336 4      (.SCB,
394    1337 4      STRING_DESC[0],
395    1338 4      STRING_DESC[SIZE]);           ! Buffer descriptor
396    1339 4
397    1340 4      INSERT_FRAME
398    1341 4      (.SCB,
399    1342 4      STRING_DESC[0],
400    1343 4      PAGE_REF[10,.FORM_LENGTH-5,.FORM_WIDTH],
401    1344 4      .FORM_WIDTH-20, 1];
402    1345 4      INSERT_FRAME
403    1346 4      (.SCB,
404    1347 4      STRING_DESC[0],
405    1348 4      PAGE_REF[14,.FORM_LENGTH-4,.FORM_WIDTH],
406    1349 4      .FORM_WIDTH-16, 1];
407    1350 4      INSERT_FRAME
408    1351 4      (.SCB,
409    1352 4      STRING_DESC[0],
410    1353 4      PAGE_REF[10,.FORM_LENGTH-3,.FORM_WIDTH],
411    1354 4      .FORM_WIDTH-20, 1];
412    1355 4      INSERT_FRAME
413    1356 4      (.SCB,
414    1357 4      STRING_DESC[0],
415    1358 4      PAGE_REF[14,.FORM_LENGTH-2,.FORM_WIDTH],
416    1359 4      .FORM_WIDTH-16, 1];
417    1360 4      INSERT_FRAME
418    1361 4      (.SCB,
419    1362 4      STRING_DESC[0],
420    1363 4      PAGE_REF[10,.FORM_LENGTH-1,.FORM_WIDTH],
421    1364 4      .FORM_WIDTH-20, 1];
422    1365 4      INSERT_FRAME
423    1366 4      (.SCB,
424    1367 4      STRING_DESC[0],
425    1368 4      PAGE_REF[14,.FORM_LENGTH,.FORM_WIDTH],
426    1369 4      .FORM_WIDTH-16, 1];
427    1370 3      END;
428    1371 2      END;
429
430    1373 2      [PSMSK_CLOSE]:          ! Return the Page of Memory
431    1374 2      RETURN_IF_ERROR_(DEALLOCATE_PAGE(.SCB));
432
433    1376 2      [OTHERWISE]:
434    1377 2      RETURN PSMS_FUNNOTSUP;
435
436    1378 2      TES; ! case .function
437
438    1381 2      SSS_NORMAL
439    1382 2
440    1383 1      END;

```

56	0000V	CF	9E	00002
5E	FDF8	CE	9E	00007
54	04	BC	D0	0000C

.ENTRY	PSMSFILE FLAG, Save R2,R3,R4,R5,R6
MOVAB	INSERT FRAME, R6
MOVAB	-520(SP) SP
MOVL	ASMB_CONTEXT, SCB

: 1251
: 1271

SEPARATE
V04-001

Print Symbiont -- separation routines
FILE_FLAG

C 14
16-Sep-1984 02:23:03 VAX-11 Bliss-32 V4.0-742
14-Sep-1984 22:32:26 [PRTSMB.SRC]SEPARATE.B32;2

Page 12
(5)

SEF
VO4

		50	0C	BC	D0	00010		MOVL	AFUNCTION, R0	1275
		05		50	D1	00014		CMPL	R0, #5	1277
		53	01FC	C4	D0	00017		BNEQ	4\$	1279
		50	026C	C4	D0	0001E		MOVL	508(SCB), PAGE_REF	1281
		01F8	C4	50	D1	00023		MOVL	620(SCB), R0	
				10	14	00028		CMPL	R0, 504(SCB)	
		51	01F8	13	0154	C4	E8	BGTR	1\$	
			C4	06	C3	0002A		BLBS	340(SCB), 2\$	1283
			51	50	D1	00035		SUBL3	#6, 504(SCB), R1	1284
				08	15	00038		CMPL	R0, R1	
				50	00000000G	8F	D0	BLEQ	2\$	
							0003A	MOVL	#PSMS_EOF, R0	1286
							1\$:	RET		
		52		10	AC	D0	00042	MOVL	FUNC_DESC, R2	1288
		62	0200	C4	D0	00046		MOVL	512(SCB), (R2)	
		50	0200	C4	C4	0004B		MULL2	512(SCB), R0	1290
		04		50	53	C1	00050	ADDL3	PAGE_REF, R0, 4(R2)	
					62	DD	00055	PUSHL	(R2)	1294
					20	DD	00057	PUSHL	#32	1293
					04	A2	DD	PUSHL	4(R2)	
		0000V	CF	03	FB	0005C		CALLS	#3, DELIMIT_STRING_NOT	
		62		50	DD	00061		MOVL	R0, (R2)	
				00FC	31	00064	3\$:	BRW	9\$	1275
		04		50	D1	00067	4\$:	CMPL	R0, #4	1298
				03	13	0006A		BEQL	5\$	
				00DC	31	0006C		BRW	7\$	
				54	DD	0006F	5\$::	PUSHL	SCB	1301
		0000V	CF	01	FB	00071		CALLS	#1, GET_FORM_SIZE	
		55	0200	C4	DO	00076		MOVL	512(SCB), FORM_WIDTH	1304
		52	01F8	C4	DO	0007B		MOVL	504(SCB), FORM_LENGTH	1305
				54	DD	00080		PUSHL	SCB	1307
		0000V	CF	01	FB	00082		CALLS	#1, ALLOCATE_PAGE	
			01	50	E8	00087		BLBS	STATUS, 6\$	
				53	04	0008A		RET		
		63	01FC	C4	D0	0008B	6\$::	MOVL	508(SCB), PAGE_REF	1309
				0C	90	00090		MOVB	#12, (PAGE_REF)	1313
				F7	A2	9F	00093	PUSHAB	-9(FORM_LENGTH)	1320
					55	DD	00096	PUSHL	FORM_WIDTH	1319
					6543	9F	00098	PUSHAB	(FORM_WIDTH)[PAGE_REF]	1318
		0000V	CF	04	FB	0009D		PUSHL	SCB	
		BD	0154	C4	E9	000A2		CALLS	#4, FILL_FILE_FLAG	1328
		6E	0200	8F	3C	000A7		BLBC	340(SCB), 3\$	1330
		04	AE	08	AE	9E	000AC	MOVZWL	#512, STRING_DESC	1331
				04	SE	DD	000B1	MOVAB	BUFFER, STRING_DESC+4	1338
					AE	9F	000B3	PUSHL	SP	1337
					54	DD	000B6	PUSHAB	STRING_DESC	1336
		0000V	CF	03	FB	000B8		PUSHL	SCB	
				01	DD	000BD		CALLS	#3, GET_VMS_LOGO	1343
		50	EC	A5	9F	000BF		PUSHAB	-20(FORM_WIDTH)	1344
		50	FB	A2	9E	000C2		MOVAB	-5(R2), R0	1343
				55	C4	000C6		MULL2	FORM_WIDTH, R0	
			OA	A043	9F	000C9		PUSHAB	10(R0)[PAGE_REF]	1342
			OC	AE	9F	000CD		PUSHAB	STRING_DESC	1343
				54	DD	000D0		PUSHL	SCB	
		66		05	FB	000D2		CALLS	#5, INSERT_FRAME	1348
				01	DD	000D5		PUSHL	#1	

SEPARATE
V04-001

Print Symbiont -- separation routines

D 14
16-Sep-1984 02:23:03 VAX-11 Bliss-32 V4.0-742
14-Sep-1984 22:32:26 [PRTSMB.SRC]SEPARATE.B32;2

Page 13
(5)

SEF
V04

50	FO	A5	9F	0000D7	PUSHAB	-16(FORM WIDTH)			1349
50	FC	A2	9E	0000DA	MOVAB	-4(R2), R0			1348
		55	C4	0000DE	MULL2	FORM WIDTH, R0			
	OE	A043	9F	0000E1	PUSHAB	14(R0)[PAGE_REF]			
	OC	AE	9F	0000E5	PUSHAB	STRING_DESC			
		54	DD	0000E8	PUSHL	SCB			
66		05	FB	0000EA	CALLS	#5, INSERT_FRAME			
		01	DD	0000ED	PUSHL	#1			
	EC	A5	9F	0000EF	PUSHAB	-20(FORM WIDTH)			
50	FD	A2	9E	0000F2	MOVAB	-3(R2), R0			
50		55	C4	0000F6	MULL2	FORM WIDTH, R0			
	OA	A043	9F	0000F9	PUSHAB	10(R0)[PAGE_REF]			
	OC	AE	9F	0000FD	PUSHAB	STRING_DESC			
		54	DD	001000	PUSHL	SCB			
66		05	FB	001002	CALLS	#5, INSERT_FRAME			
		01	DD	001005	PUSHL	#1			
	FO	A5	9F	001007	PUSHAB	-16(FORM WIDTH)			
50	FE	A2	9E	0010A0	MOVAB	-2(R2), R0			
50		55	C4	0010E0	MULL2	FORM WIDTH, R0			
	OE	A043	9F	001111	PUSHAB	14(R0)[PAGE_REF]			
	OC	AE	9F	001115	PUSHAB	STRING_DESC			
		54	DD	001118	PUSHL	SCB			
66		05	FB	0011A0	CALLS	#5, INSERT_FRAME			
		01	DD	0011D0	PUSHL	#1			
	EC	A5	9F	0011F0	PUSHAB	-20(FORM WIDTH)			
50	FF	A2	9E	001220	MOVAB	-1(R2), R0			
50		55	C4	001260	MULL2	FORM WIDTH, R0			
	OA	A043	9F	001290	PUSHAB	10(R0)[PAGE_REF]			
	OC	AE	9F	0012D0	PUSHAB	STRING_DESC			
		54	DD	001300	PUSHL	SCB			
66		05	FB	001320	CALLS	#5, INSERT_FRAME			
		01	DD	001350	PUSHL	#1			
	FO	A5	9F	001370	PUSHAB	-16(FORM WIDTH)			
52		55	C4	0013A0	MULL2	FORM WIDTH, R2			
	OE	A243	9F	0013D0	PUSHAB	14(R2)[PAGE_REF]			
	OC	AE	9F	001410	PUSHAB	STRING_DESC			
		54	DD	001440	PUSHL	SCB			
66		05	FB	001460	CALLS	#5, INSERT_FRAME			
		18	11	001490	BRB	9\$			
02		50	D1	0014B0	7\$:	CMPL	R0, #2		
		08	12	0014E0	BNEQ	8\$			
		54	DD	001500	PUSHL	SCB			
0000V	CF	01	FB	001520	CALLS	#1, DEALLOCATE_PAGE			
09		50	F8	001570	BLBS	STATUS, 9\$			
		14	0015A0		RET				
50	00000000G	8F		0015B0	8\$:	MOVL	#PSMS_FUNNOTSUP, R0		
		04	001620		RET				
50		01	DD	001630	9\$:	MOVL	#1, R0		
		04	001660		RET				

; Routine Size: 359 bytes, Routine Base: CODE + 00B1

```
442 1384 1 %sbttl 'PSMSFILE_TRAILER - Print a File Trailer Page'
443 1385 1 Functional Description:
444 1386 1 This routine controls the creation of the file flag page. The
445 1387 1 FUNCTION code dictates the action taken in creation.
446 1388 1 FUNCTION:
447 1389 1 OPEN - Allocate and create the file Flag Page
448 1390 1 READ - Return the current line of the file Flag Page
449 1391 1 CLOSE - Return the buffer allocated on OPEN
450 1392 1
451 1393 1 Formal Parameters:
452 1394 1 SMB_CONTEXT - Pointer to the SMB
453 1395 1 USER_CONTEXT - User defined pointer (not used here)
454 1396 1 FUNCTION - OPEN, READ, CLOSE
455 1397 1 FUNC_DESC - Pointer to functionally dependent descriptor
456 1398 1 FUNC_ARG - Pointer to functionally dependent argument
457 1399 1
458 1400 1 Implicit Inputs:
459 1401 1 none
460 1402 1
461 1403 1 Implicit Outputs:
462 1404 1 none
463 1405 1
464 1406 1 Returned Value:
465 1407 1 none
466 1408 1
467 1409 1 Side Effects:
468 1410 1 none
469 1411 1 --.
470 1412 1
471 1413 1 GLOBAL ROUTINE PSMSFILE_TRAILER ( %SBTTL 'FILE_TRAILER'
472 1414 1     SMB_CONTEXT : REF VECTOR,
473 1415 1     USER_CONTEXT : REF VECTOR,
474 1416 1     FUNCTION : REF VECTOR,
475 1417 1     FUNC_DESC : REF VECTOR,
476 1418 1     FUNC_ARG : REF VECTOR
477 1419 1 ) =
478 1420 2 BEGIN
479 1421 2
480 1422 2 LITERAL
481 1423 2 TRAILING = 1;
482 1424 2 LOCAL
483 1425 2 SCB : REF $BLOCK,
484 1426 2 STATUS,
485 1427 2 FORM_WIDTH,
486 1428 2 FORM_LENGTH,
487 1429 2 FORM_SIZE,
488 1430 2 PAGE_REF : REF PAGE_ARRAY; ! Declare the pointer
489 1431 2 ! to page
490 1432 2
491 1433 2 SCB = .SMB_CONTEXT[0];
492 1434 2
493 1435 2 ! Check the FUNCTION requested
494 1436 2
495 1437 2 SELECTONEU .FUNCTION[0] OF
496 1438 2 SET
497 1439 2 [PSMSK READ]:
498 1440 3 BEGIN
```

Print Symbiont -- separation routines
FILE_TRAILER

F 14
16-Sep-1984 02:23:03
14-Sep-1984 22:32:26
VAX-11 Bliss-32 V4.0-742
[PRTSMB.SRC]SEPARATE.B32:2

1441 3 LOCAL TEMP_PTR;
1442 3
1443 3 PAGE_REF = .SCB[PSMSA_PAGE_POINTER];
1444 3
1445 3 IF .SCB[PSMSL_RECORD_NUMBER] GTR (.SCB[PSMSL_PAGE_LENGTH] - 6 - 2) THEN
1446 3 RETURN PSMS_EOF;
1447 3
1448 3 FUNC_DESC[SIZE] = .SCB[PSMSL_PAGE_WIDTH];
1449 3
1450 3 FUNC_DESC[ADDR] = PAGE_REF[0,.SCB[PSMSL_RECORD_NUMBER],
1451 3 .SCB[PSMSL_PAGE_WIDTH]];
1452 3
1453 3 FUNC_DESC[SIZE] = DELIMIT_STRING_NOT (.FUNC_DESC[ADDR],
1454 3 %CHAR732), .FUNC_DESC[SIZE]);
1455 3
1456 2 END;
1457 2
1458 2 [PSMSK_OPEN]:
1459 3 BEGIN
1460 3
1461 3 GET_FORM_SIZE (.SCB); ! Returns the WidthxLength
1462 3
1463 3
1464 3 FORM_WIDTH = .SCB[PSMSL_PAGE_WIDTH];
1465 3 FORM_LENGTH = .SCB[PSMSL_PAGE_LENGTH];
1466 3
1467 3 RETURN_IF_ERROR_(ALLOCATE_PAGE(.SCB)); ! Get the page of memory
1468 3
1469 3 PAGE_REF = .SCB[PSMSA_PAGE_POINTER]; ! My local page pointer
1470 3
1471 3 ! Always start at top of page
1472 3
1473 3 PAGE_REF[0,0,.FORM_WIDTH] = PSMSK_CHAR_FF; ! form feed in 0 pos.
1474 3
1475 3 ! Standard Trailer Page 132x66: text covers rows 1 through 58,
1476 3 translated to frames... ref starts at 1 and length is 57.
1477 3 FILL_FILE_TRAILER(.SCB,
1478 3 PAGE_REF[0,1,.FORM_WIDTH],
1479 3 .FORM_WIDTH,
1480 3 .FORM_LENGTH - 6 - 2 - 1); ! total form length...
1481 2 ! ..6 burst, 2 spaces
1482 2 ! ...top margin of 1
1483 2
1484 2 [PSMSK_CLOSE]: ! Return the Page of Memory
1485 2 RETURN_IF_ERROR_(DEALLOCATE_PAGE(.SCB));
1486 2
1487 2 [OTHERWISE]:
1488 2 RETURN PSMS_FUNNOTSUP;
1489 2
1490 2 TES: ! case .function
1491 2
1492 2 SSS_NORMAL
1493 2
1494 1 END;

SEPARATE
V04-001

Print_Symbiont -- separation routines
FILE_TRAILER

G 14
16-Sep-1984 02:23:03
14-Sep-1984 22:32:26

VAX-11 Bliss-32 V4.0-742
[PRTSMB.SRC]SEPARATE.B32;2

Page 16
(6)

SE
VO

				003C	00000	.ENTRY	PSMSFILE TRAILER, Save R2,R3,R4,R5	1413	
		52	04	BC	00002	MOVL	ASMB_CONTEXT, SCB	1433	
		50	0C	BC	00006	MOVL	FUNCTION, R0	1437	
		05		50	D1 0000A	CMPL	R0, #5	1439	
				41	12 0000D	BNEQ	28		
50	01F8	54	01FC	C2	0000F	MOVL	508(SCB), PAGE_REF	1443	
		C2		08	C3 00014	SUBL3	#8, 504(SCB), R0	1445	
		50	026C	C2	D1 0001A	CMPL	620(SCB), R0		
				08	15 0001F	BLEQ	18		
		50	00000000G	8F	00021	MOVL	#PSMS_EOF, R0	1446	
					04 00028	RET			
						MOVL	FUNC_DESC, R3	1448	
04	A3	53	10	AC	00029	18:	512(SCB), (R3)		
		63	0200	C2	0002D	MOVL	512(SCB), 620(SCB), R0	1450	
		C2	0200	C2	C5 00032	MULL3	PAGE_REF, R0, 4(R3)	1454	
		50		54	C1 0003A	ADDL3	(R3)	1453	
				63	DD 0003F	PUSHL	#32		
				20	DD 00041	PUSHL	4(R3)		
		0000V	04	A3	DD 00043	PUSHL	#3, DELIMIT_STRING_NOT		
		CF		03	FB 00046	CALLS	RO, (R3)		
		63		50	00048	MOVL	SS	1437	
				51	11 0004E	BRB	58	1458	
		04		50	D1 00050	28:	CMPL	RO, #4	
				34	12 00053	BNEQ	38		
		0000V		52	DD 00055	PUSHL	SCB	1461	
		CF		01	FB 00057	CALLS	#1, GET_FORM_SIZE		
		55	0200	C2	0005C	MOVL	512(SCB), FORM_WIDTH	1464	
		53	01F8	C2	00061	MOVL	504(SCB), FORM_LENGTH	1465	
				52	DD 00066	PUSHL	SCB	1467	
		0000V	CF	01	FB 00068	CALLS	#1, ALLOCATE_PAGE		
		34		50	E9 0006D	BLBC	STATUS, 68		
		54	01FC	C2	00070	MOVL	508(SCB), PAGE_REF	1469	
		64		0C	90 00075	MOVB	#12, (PAGE_REF)	1473	
			F7	A3	9F 00078	PUSHAB	-9(FORM_LENGTH)	1480	
				55	DD 00078	PUSHL	FORM_WIDTH	1479	
				6544	9F 0007D	PUSHAB	(FORM_WIDTH)[PAGE_REF]	1478	
		0000V	CF	52	DD 00080	PUSHL	SCB		
				04	FB 00082	CALLS	#4, FILL_FILE_TRAILER		
				18	11 00087	BRB	58	1437	
		02		50	D1 00089	38:	CMPL	RO, #2	1484
				0B	12 0008C	BNEQ	48		
		0000V	CF	52	DD 0008E	PUSHL	SCB	1485	
		09		01	FB 00090	CALLS	#1, DEALLOCATE_PAGE		
				50	E8 00095	BLBS	STATUS, 58		
		50	00000000G	8F	04 00098	RET			
					04 000A0	MOVL	#PSMS_FUNNOTSUP, R0	1488	
		50		01	000A1	58:	RET		
				04	000A4	68:	MOVL	#1, R0	1494
						RET			

; Routine Size: 165 bytes, Routine Base: CODE + 0218

```
554 1495 1 %sbttl 'PSMSJOB_BURST - Print a Job Burst Page'  
555 1496 1 Functional Description:  
556 1497 1 This routine controls the creation of the job burst page. The  
557 1498 1 FUNCTION code dictates the action taken in creation.  
558 1499 1  
559 1500 1 FUNCTION:  
560 1501 1 OPEN - Allocate and create the Job Burst Page  
561 1502 1 READ - Return the current line of the Job Burst Page  
562 1503 1 CLOSE - Return the buffer allocated on OPEN  
563 1504 1 Formal Parameters:  
564 1505 1 SMB_CONTEXT - Pointer to the SMB  
565 1506 1 USER_CONTEXT - User defined pointer (not used here)  
566 1507 1 FUNCTION - OPEN, READ, CLOSE  
567 1508 1 FUNC_DESC - Pointer to functionally dependent descriptor  
568 1509 1 FUNC_ARG - Pointer to functionally dependent argument  
569 1510 1  
570 1511 1 Implicit Inputs:  
571 1512 1 none  
572 1513 1  
573 1514 1 Implicit Outputs:  
574 1515 1 none  
575 1516 1  
576 1517 1 Returned Value:  
577 1518 1 none  
578 1519 1  
579 1520 1 Side Effects:  
580 1521 1 none  
581 1522 1--  
582 1523 1 GLOBAL ROUTINE PSMSJOB_BURST ( %SBTTL 'JOB_BURST'  
583 1524 1 SMB_CONTEXT : REF VECTOR,  
584 1525 1 USER_CONTEXT : REF VECTOR,  
585 1526 1 FUNCTION : REF VECTOR,  
586 1527 1 FUNC_DESC : REF VECTOR,  
587 1528 1 FUNC_ARG : REF VECTOR  
588 1529 1 ) =  
589 1530 2 BEGIN  
590 1531 2 LOCAL  
591 1532 2 SCB : REF $BBLOCK,  
592 1533 2 STATUS,  
593 1534 2 FORM_WIDTH,  
594 1535 2 FORM_LENGTH,  
595 1536 2 FORM_SIZE,  
596 1537 2 PAGE_REF : REF PAGE_ARRAY; ! Declare the pointer  
597 1538 2 ! to page  
598 1539 2  
599 1540 2 SCB = .SMB_CONTEXT[0];  
600 1541 2  
601 1542 2 ! Check the FUNCTION requested  
602 1543 2  
603 1544 2 SELECTONEU .FUNCTION[0] OF  
604 1545 2 SET  
605 1546 2 [PSMSK READ]:  
606 1547 2 BEGIN  
607 1548 2 PAGE_REF = .SCB[PSMSA_PAGE_POINTER]; ! Output one line at a time  
608 1549 2  
609 1550 3 IF .SCB[PSMSL_RECORD_NUMBER] GTR (.SCB[PSMSL_PAGE_LENGTH] - 6)  
610 1551 4
```

SEPARATE
V04-001

Print Symbiont -- separation routines
JOB_BURST

14
16-Sep-1984 02:23:03
14-Sep-1984 22:32:26
VAX-11 Bliss-32 V4.0-742
[PRTSMB.SRC]SEPARATE.B32;2

Page 18
(7)

SE
VC

611 1552 3 THEN
612 1553 RETURN PSMS_EOF;
613 1554
614 1555 FUNC_DESC[SIZE] = .SCB[PSMSL_PAGE_WIDTH];
615 1556 FUNC_DESC[ADDR] = PAGE_REF[0].SCB[PSMSL_RECORD_NUMBER].
616 1557 .SCB[PSMSL_PAGE_WIDTH];
617 1558
618 1559 ! adjust pointer
619 1560 FUNC_DESC[SIZE] = DELIMIT_STRING_NOT (.FUNC_DESC[ADDR],
620 1561 .CHAR[32], .FUNC_DESC[SIZE]);
621 1562
622 1563 END;
623 1564
624 1565 [PSMSK_OPEN]:
625 1566 BEGIN
626 1567 GET_FORM_SIZE (.SCB); ! Returns the WidthxLength
627 1568
628 1569
629 1570
630 1571 FORM_WIDTH = .SCB[PSMSL_PAGE_WIDTH];
631 1572 FORM_LENGTH = .SCB[PSMSL_PAGE_LENGTH];
632 1573
633 1574 RETURN_IF_ERROR_(ALLOCATE_PAGE(.SCB)); ! Get the page of memory
634 1575
635 1576 PAGE_REF = .SCB[PSMSA_PAGE_POINTER]; ! My local page pointer
636 1577
637 1578 ! No form_feed on a burst page
638 1579
639 1580 ! Standard Burst Page 132x66: text covers rows 2 through 60,
640 1581 ! translated to frames... ref starts at 2 and length is 58.
641 1582 FILL_JOB_FLAG(.SCB,
642 1583 PAGE_REF[0,2,.FORM_WIDTH],
643 1584 .FORM_WIDTH,
644 1585 .FORM_LENGTH - 6 - 2); ! 6 blank lines
645 1586 ! top margin is 2
646 1587
647 1588 [PSMSK_CLOSE]: ! Return the Page of Memory
648 1589 RETURN_IF_ERROR_(DEALLOCATE_PAGE(.SCB));
649 1590
650 1591 [OTHERWISE]:
651 1592 RETURN PSMS_FUNNOTSUP;
652 1593
653 1594 TES; ! case .function
654 1595
655 1596 SSS_NORMAL
656 1597
657 1598 1 END;

52	04	003C 00000	.ENTRY	PSMSJOB_BURST, Save R2,R3,R4,R5	: 1523
50	0C	BC D0 00002	MOVL	ASMB_CONTEXT, SCB	: 1541
05	50	BC D0 00006	MOVL	FUNCTION, R0	: 1545
	41	D1 0000A	CMPL	R0, #5	: 1547
		41 12 0000D	BNEQ	2\$	

; Routine Size: 162 bytes, Routine Base: CODE + 02BD

659 1599 1 %sbttl 'PSMS\$JOB_FLAG - Print a Job Flag Page'
660 1600 1 Functional Description:
661 1601 1 This routine controls the creation of the job flag page. The
662 1602 1 FUNCTION code dictates the action taken in creation.
663 1603 1 FUNCTION:
664 1604 1 OPEN - Allocate and create the Job Flag Page
665 1605 1 READ - Return the current line of the Job Flag Page
666 1606 1 CLOSE - Return the buffer allocated on OPEN
667 1607 1
668 1608 1 Formal Parameters:
669 1609 1 SMB_CONTEXT - Pointer to the SMB
670 1610 1 USER_CONTEXT - User defined pointer (not used here)
671 1611 1 FUNCTION - OPEN, READ, CLOSE
672 1612 1 FUNC_DESC - Pointer to functionally dependent descriptor
673 1613 1 FUNC_ARG - Pointer to functionally dependent argument
674 1614 1
675 1615 1 Implicit Inputs:
676 1616 1 none
677 1617 1
678 1618 1 Implicit Outputs:
679 1619 1 none
680 1620 1
681 1621 1 Returned Value:
682 1622 1 none
683 1623 1
684 1624 1 Side Effects:
685 1625 1 none
686 1626 1 --
687 1627 1 GLOBAL ROUTINE PSMS\$JOB_FLAG (%SBTTL 'JOB_FLAG'
688 1628 1 SMB_CONTEXT : REF VECTOR,
689 1629 1 USER_CONTEXT : REF VECTOR,
690 1630 1 FUNCTION : REF VECTOR,
691 1631 1 FUNC_DESC : REF VECTOR,
692 1632 1 FUNC_ARG : REF VECTOR
693 1633 1) =
694 1634 2 BEGIN
695 1635 2 LITERAL
696 1636 2 TRAILING = 1;
697 1637 2
698 1638 2 LOCAL
699 1639 2 SCB : REF SBBLOCK,
700 1640 2 STATUS,
701 1641 2 FORM_WIDTH,
702 1642 2 FORM_LENGTH,
703 1643 2 FORM_SIZE,
704 1644 2 BUFFER : VECTOR [512,byte], ! Assume max size 512 bytes
705 1645 2 STRING_DESC : VECTOR [2], ! Descriptor to current string
706 1646 2 PAGE_REF : REF PAGE_ARRAY; ! Declare the pointer
707 1647 2
708 1648 2
709 1649 2 SCB = .SMB_CONTEXT[0];
710 1650 2
711 1651 2 ! Check the FUNCTION requested
712 1652 2
713 1653 2 SELECTONEU .FUNCTION[0] OF
714 1654 2 SET
715 1655 2 [PSMSK_READ]:

SEPARATE
V04-001

Print Symbiont -- separation routines
JOB_FLAG

L 14
16-Sep-1984 02:23:03 VAX-11 Bliss-32 v4.0-742
14-Sep-1984 22:32:26 [PRTSMB.SRC]SEPARATE.B32;2

Page 21
(8)

```
716    1656 3      BEGIN
717    1657 3      PAGE_REF = .SCB[PSMSA_PAGE_POINTER];
718    1658 3      ! Output one line at a time
719    1659 4      IF ( .SCB[PSMSL_RECORD_NUMBER] GTR .SCB[PSMSL_PAGE_LENGTH])
720    1660 3          OR
721    1661 4          ( NOT .SEPARATE FLAG (JOB_BURST) AND .SCB[PSMSL_RECORD_NUMBER]
722    1662 4          GEQ (.SCB[PSMSL_PAGE_LENGTH] - 6 ) )
723    1663 3          THEN
724    1664 3          RETURN PSMS_EOF;
725    1665 3
726    1666 3      FUNC_DESC[SIZE] = .SCB[PSMSL_PAGE_WIDTH];
727    1667 3      FUNC_DESC[ADDR] = PAGE_REF[0..SCB[PSMSL_RECORD_NUMBER],
728    1668 3          .SCB[PSMSL_PAGE_WIDTH]];
729    1669 3
730    1670 3      ! adjust pointer
731    1671 3      FUNC_DESC[SIZE] = DELIMIT_STRING_NOT (.FUNC_DESC[ADDR],
732    1672 3          %CHAR(32), .FUNC_DESC[SIZE]);
733    1673 3
734    1674 2      END;
735    1675 2
736    1676 2      [PSMSK OPEN]:
737    1677 3      BEGIN
738    1678 3
739    1679 3      GET_FORM_SIZE (.SCB);                      ! Returns the WidthxLength
740    1680 3
741    1681 3
742    1682 3      FORM_WIDTH      = .SCB[PSMSL_PAGE_WIDTH];
743    1683 3      FORM_LENGTH     = .SCB[PSMSL_PAGE_LENGTH];
744    1684 3
745    1685 3      RETURN_IF_ERROR_(ALLOCATE_PAGE( .SCB));   ! Get the page of memory
746    1686 3
747    1687 3      PAGE_REF = .SCB[PSMSA_PAGE_POINTER];       ! My local page pointer
748    1688 3
749    1689 3      ! Always start at top of page
750    1690 3
751    1691 3      PAGE_REF[0,0..FORM_WIDTH] = PSMSK_CHAR_FF;   ! form feed in 0 pos.
752    1692 3
753    1693 3      ! Standard Flag Page 132x66: text covers rows 1 through 58
754    1694 3      ! translated to frames... ref starts at 1 and length is 57.
755    1695 3      FILL_JOB_FLAG( .SCB,
756    1696 3          PAGE_REF[0,1..FORM_WIDTH],
757    1697 3          :FORM_WIDTH,
758    1698 3          :FORM_LENGTH - 6 - 2 - 1);           ! 6 burst,
759    1699 3          ! 2 spaces before burst,
760    1700 3          ! top margin is 1
761    1701 4      IF (.SEPARATE_FLAG_(JOB_BURST))
762    1702 3          THEN
763    1703 4          BEGIN
764    1704 4          STRING_DESC[SIZE] = %ALLOCATION(BUFFER);
765    1705 4          STRING_DESC[ADDR] = BUFFER;
766    1706 4
767    1707 4      GET_VMS_LOGO
768    1708 4          (.SCB,
769    1709 4          STRING_DESC[0],
770    1710 4          STRING_DESC[SIZE]);                  ! Buffer descriptor
771    1711 4
772    1712 4      INSERT_FRAME
```

SEPARATE
V04-001Print Symbiont -- separation routines
JOB_FLAGM 14
16-Sep-1984 02:23:03
14-Sep-1984 22:32:26
VAX-11 Bliss-32 V4.0-742
[PRTSMB.SRC]SEPARATE.B32;2Page 22
(8)

```

773    1713 4      (.SCB,
774    1714 4      STRING_DESC[0],
775    1715 4      PAGE_REF[10,,FORM_LENGTH-5,,FORM_WIDTH],
776    1716 4      .FORM_WIDTH-20, 1T;
777    1717 4      INSERT_FRAME
778    1718 4      (.SCB,
779    1719 4      STRING_DESC[0],
780    1720 4      PAGE_REF[14,,FORM_LENGTH-4,,FORM_WIDTH],
781    1721 4      .FORM_WIDTH-16, 1T;
782    1722 4      INSERT_FRAME
783    1723 4      (.SCB,
784    1724 4      STRING_DESC[0],
785    1725 4      PAGE_REF[10,,FORM_LENGTH-3,,FORM_WIDTH],
786    1726 4      .FORM_WIDTH-20, 1T;
787    1727 4      INSERT_FRAME
788    1728 4      (.SCB,
789    1729 4      STRING_DESC[0],
790    1730 4      PAGE_REF[14,,FORM_LENGTH-2,,FORM_WIDTH],
791    1731 4      .FORM_WIDTH-16, 1T;
792    1732 4      INSERT_FRAME
793    1733 4      (.SCB,
794    1734 4      STRING_DESC[0],
795    1735 4      PAGE_REF[10,,FORM_LENGTH-1,,FORM_WIDTH],
796    1736 4      .FORM_WIDTH-20, 1T;
797    1737 4      INSERT_FRAME
798    1738 4      (.SCB,
799    1739 4      STRING_DESC[0],
800    1740 4      PAGE_REF[14,,FORM_LENGTH,,FORM_WIDTH],
801    1741 4      .FORM_WIDTH-16, 1T;
802    1742 3      END:
803    1743 2      END;
804    1744 2
805    1745 2      [PSMSK_CLOSE]:           ! Return the Page of Memory
806    1746 2      RETURN_IF_ERROR_(DEALLOCATE_PAGE(.SCB));
807    1747 2
808    1748 2      [OTHERWISE]:
809    1749 2      RETURN PSMS_FUNNOTSUP;
810    1750 2
811    1751 2      TES; ! case .function
812    1752 2
813    1753 2      SSS_NORMAL
814    1754 2
815    1755 1      END;

```

56	0000V	007C	00000	.ENTRY	PSMSJOB FLAG, Save R2,R3,R4,R5,R6	: 1627
5E	FDF8	CE	9E 00002	MOVAB	INSERT FRAME, R6	
54	04	BC	D0 00007	MOVAB	-520(SP), SP	: 1649
50	0C	BC	D0 0000C	MOVL	ASMB_CONTEXT, SCB	: 1653
05		50	D1 00010	MOVL	FUNCTION, R6	: 1655
		4F	12 00014	CMPL	R0, #5	
53	01FC	C4	D0 00019	BNEQ	4\$	
50	026C	C4	D0 0001E	MOVL	508(SCB), PAGE_REF	: 1657
				MOVL	620(SCB), R0	: 1659

**SEPARATE
V04-001**

Print Symbiont -- separation routines
JOB_FLAG

N 14
16-Sep-1984 02:23:03 VAX-11 Bliss-32 V4.0-742
14-Sep-1984 22:32:26 [PRTSMB.SRC]SEPARATE.832;2

Page 23
(8)

	01F8	C4	50	D1	00023	CMPL	R0, 504(SCB)
13	0154	C4	11	14	00028	BGTR	1\$
51	01F8	C4	05	E0	0002A	BBS	#5, 340(SCB), 2\$
		51	06	C3	00030	SUGL3	#6, 504(SCB), R1
			50	D1	00036	CMPL	R0, R1
			08	19	00039	BLSS	2\$
	50	00000000G	8F	D0	0003B	1\$:	MOVL #PSMS_EOF, R0
				04	00042	RET	
04	A2	52	10	AC	D0	00043	2\$:
		62	0200	C4	D0	00047	MOVL FUNC_DESC, R2
		50	0200	C4	C4	0004C	512(SCB), (R2)
				53	C1	00051	512(SCB), R0
				62	DD	00056	PULL2 PAGE_REF, R0, 4(R2)
				20	DD	00058	(R2)
				A2	DD	0005A	PUSHL #32
	0000V	CF	04	03	FB	0005D	PUSHL 4(R2)
		62		50	D0	00062	CALLS #3, DELIMIT_STRING_NOT
			00FD	31	00065	MOVL R0, (R2)	
		04		50	D1	00068	3\$:
				03	13	0006B	CMPL R0, #4
			00DD	31	0006D	BEQL 5\$	
				54	DD	00070	BRW 7\$
	0000V	CF		01	FB	00072	PUSHL SCB
		55	0200	C4	D0	00077	CALLS #1, GET_FORM_SIZE
		52	01F8	C4	D0	0007C	MOVL 512(SCB), FORM_WIDTH
	0000V	CF		54	DD	00081	MOVL 504(SCB), FORM_LENGTH
		01		01	FB	00083	PUSHL SCB
				50	E8	00088	CALLS #1, ALLOCATE_PAGE
				04	0008B	BLBS STATUS, 6\$	
		53	01FC	C4	D0	0008C	RET
		63		0C	90	00091	MOVL 508(SCB), PAGE_REF
			F7	A2	9F	00094	MOVB #12, (PAGE_REF)
				55	DD	00097	-9(FORM_LENGTH)
			6543	9F	00099	PUSHAB FORM_WIDTH	
				54	DD	0009C	PUSHAB (FORM_WIDTH)[PAGE_REF]
	0000V	CF		04	FB	0009E	PUSHL SCB
BC	0154	C4		05	E1	000A3	CALLS #4, FILL_JOB_FLAG
		6E	0200	8F	3C	000A9	BBB #5, 340(SCB), 3\$
	04	AE	08	AE	9E	000AE	MOVZWL #512, STRING_DESC
				5E	DD	000B3	MOVAB BUFFER, STRING_DESC+4
			04	AE	9F	000B5	PUSHL SP
	0000V	CF		54	DD	000B8	PUSHAB STRING_DESC
				03	FB	000BA	PUSHL SCB
				01	DD	000BF	CALLS #3, GET_VMS_LOGO
		50	EC	A5	9F	000C1	PUSHAB -20(FORM_WIDTH)
			FB	A2	9E	000C4	MOVAB -5(R2), R0
		50		55	C4	000C8	MULL2 FORM_WIDTH, R0
			0A	A043	9F	000CB	PUSHAB 10(R0)[PAGE_REF]
			OC	AE	9F	000CF	PUSHAB STRING_DESC
				54	DD	000D2	PUSHL SCB
		66		05	FB	000D4	CALLS #5, INSERT_FRAME
				01	DD	000D7	PUSHL #1
		50	F0	A5	9F	000D9	PUSHAB -16(FORM_WIDTH)
			FC	A2	9E	000DC	MOVAB -4(R2), R0
		50		55	C4	000E0	MULL2 FORM_WIDTH, R0
			OE	A043	9F	000E3	PUSHAB 14(R0)[PAGE_REF]
		50	OC	AE	9F	000E7	PUSHAB STRING_DESC

SEPARATE
V04-001

Print Symbiont -- separation routines
JOB_FLAG

B 15
16-Sep-1984 02:23:03
14-Sep-1984 22:32:26

VAX-11 Bliss-32 V4.0-742
[PRTSMB.SRC]SEPARATE.B32;2

Page 24
(8)

66		54 DD 000EA	PUSHL	SCB			1720
		05 FB 000EC	CALLS	#5, INSERT_FRAME			1725
		01 DD 000EF	PUSHL	#1			1726
50	EC	A5 9F 000F1	PUSHAB	-20(FORM WIDTH)			1725
50	FD	A2 9E 000F4	MOVAB	-3(R2), R0			1724
		55 C4 000F8	MULL2	FORM WIDTH, R0			1725
	OA	A043 9F 000FB	PUSHAB	10(R0)[PAGE_REF]			1724
	OC	AE 9F 000FF	PUSHAB	STRING_DESC			1725
		54 DD 00102	PUSHL	SCB			1724
66		05 FB 00104	CALLS	#5, INSERT_FRAME			1725
		01 DD 00107	PUSHL	#1			1730
50	FO	A5 9F 00109	PUSHAB	-16(FORM WIDTH)			1731
50	FE	A2 9E 0010C	MOVAB	-2(R2), R0			1730
		55 C4 00110	MULL2	FORM WIDTH, R0			1729
	OE	A043 9F 00113	PUSHAB	14(R0)[PAGE_REF]			1730
	OC	AE 9F 00117	PUSHAB	STRING_DESC			1729
		54 DD 0011A	PUSHL	SCB			1730
66		05 FB 0011C	CALLS	#5, INSERT_FRAME			1735
		01 DD 0011F	PUSHL	#1			1735
50	EC	A5 9F 00121	PUSHAB	-20(FORM WIDTH)			1736
50	FF	A2 9E 00124	MOVAB	-1(R2), R0			1735
		55 C4 00128	MULL2	FORM WIDTH, R0			1734
	OA	A043 9F 0012B	PUSHAB	10(R0)[PAGE_REF]			1735
	OC	AE 9F 0012F	PUSHAB	STRING_DESC			1734
		54 DD 00132	PUSHL	SCB			1735
66		05 FB 00134	CALLS	#5, INSERT_FRAME			1740
		01 DD 00137	PUSHI	#1			1741
52	FO	A5 9F 00139	PUSHAB	-16(FORM WIDTH)			1741
		55 C4 0013C	MULL2	FORM WIDTH, R2			1740
	OE	A243 9F 0013F	PUSHAB	14(R2)[PAGE_REF]			1739
	OC	AE 9F 00143	PUSHAB	STRING_DESC			1740
		54 DD 00146	PUSHL	SCB			1740
66		05 FB 00148	CALLS	#5, INSERT_FRAME			1653
		18 11 0014B	BRB	9\$			1745
02		50 D1 0014D	7\$:	CMPL	R0, #2		1745
		0B 12 00150	BNEQ	8\$			1746
		54 DD 00152	PUSHL	SCB			1746
0000V	CF	01 FB 00154	CALLS	#1, DEALLOCATE_PAGE			
09		50 E8 00159	BLBS	STATUS, 9\$			
		04 0015C	RET				
50	00000000G	8F D0 0015D	8\$:	NPMS_FUNNOTSUP, R0			1749
		04 00164	RET				
50		01 D0 00165	9\$:	NPMS_FUNNOTSUP, R0			1755
		04 00168	RET				

; Routine Size: 361 bytes, Routine Base: CODE + 035F

```
817 1756 1 %sbttl 'PSMSJOB TRAILER - Print a Job Flag Page'
818 1757 1 Functional Description:
819 1758 1 This routine controls the creation of the job trailer page. The
820 1759 1 FUNCTION code dictates the action taken in creation.
821 1760 1 FUNCTION:
822 1761 1 OPEN - Allocate and create the Job Trailer Page
823 1762 1 READ - Return the current line of the Job Trailer Page
824 1763 1 CLOSE - Return the buffer allocated on OPEN
825 1764 1
826 1765 1 Formal Parameters:
827 1766 1 SMB_CONTEXT - Pointer to the SMB
828 1767 1 USER_CONTEXT - User defined pointer (not used here)
829 1768 1 FUNCTION - OPEN, READ, CLOSE
830 1769 1 FUNC_DESC - Pointer to functionally dependent descriptor
831 1770 1 FUNC_ARG - Pointer to functionally dependent argument
832 1771 1
833 1772 1 Implicit Inputs:
834 1773 1 none
835 1774 1
836 1775 1 Implicit Outputs:
837 1776 1 none
838 1777 1
839 1778 1 Returned Value:
840 1779 1 none
841 1780 1
842 1781 1 Side Effects:
843 1782 1 none
844 1783 1 --
845 1784 1 GLOBAL ROUTINE PSMSJOB_TRAILER ( %SBTTL 'JOB_TRAILER'
846 1785 1 SMB_CONTEXT : REF VECTOR,
847 1786 1 USER_CONTEXT : REF VECTOR,
848 1787 1 FUNCTION : REF VECTOR,
849 1788 1 FUNC_DESC : REF VECTOR,
850 1789 1 FUNC_ARG : REF VECTOR
851 1790 1 ) =
852 1791 2 BEGIN
853 1792 2
854 1793 2 LITERAL
855 1794 2 TRAILING = 1;
856 1795 2 LOCAL
857 1796 2 SCB : REF $BBLOCK,
858 1797 2 STATUS,
859 1798 2 FORM_WIDTH,
860 1799 2 FORM_LENGTH,
861 1800 2 FORM_SIZE,
862 1801 2 PAGE_REF : REF PAGE_ARRAY; ! Declare the pointer
863 1802 2 ! to page
864 1803 2
865 1804 2 SCB = .SMB_CONTEXT[0];
866 1805 2
867 1806 2 ! Check the FUNCTION requested
868 1807 2
869 1808 2 SELECTONEU .FUNCTION[0] OF
870 1809 2 SET
871 1810 2 [PSMSK_READ]:
872 1811 2 BEGIN
873 1812 3 LOCAL TEMP_PTR;
```

874 1813 3
875 1814 3
876 1815 4
877 1816 4
878 1817 4
879 1818 4
880 1819 4
881 1820 4
882 1821 4
883 1822 4
884 1823 4
885 1824 4
886 1825 4
887 1826 4
888 1827 4
889 1828 4
890 1829 4
891 1830 4
892 1831 4
893 1832 4
894 1833 4
895 1834 4
896 1835 4
897 1836 4
898 1837 4
899 1838 4
900 1839 4
901 1840 4
902 1841 4
903 1842 4
904 1843 4
905 1844 4
906 1845 4
907 1846 4
908 1847 4
909 1848 4
910 1849 4
911 1850 4
912 1851 4
913 1852 4
914 1853 4
915 1854 4
916 1855 2
917 1856 2
918 1857 2
919 1858 2
920 1859 2
921 1860 2
922 1861 2
923 1862 2
924 1863 2
925 1864 2
926 1865 1

3 PAGE_REF = .SCB[PSMSA_PAGE_POINTER];
4 IF .SCB[PSMSL_RECORD_NUMBER] GTR (.SCB[PSMSL_PAGE_LENGTH] - 2 - 6)
5 THEN
6 RETURN PSMS_EOF;
7 FUNC_DESC[SIZE] = .SCB[PSMSL_PAGE_WIDTH];
8 FUNC_DESC[ADDR] = PAGE_REF[0,.SCB[PSMSL_RECORD_NUMBER],
9 .SCB[PSMSL_PAGE_WIDTH]];
10 FUNC_DESC[SIZE] = DELIMIT_STRING_NOT (.FUNC_DESC[ADDR],
11 ! adjust pointer
12 %CHAR732), .FUNC_DESC[SIZE]);
13 END;
14 [PSMSK OPEN]:
15 BEGIN
16 GET_FORM_SIZE (.SCB); ! Returns the WidthxLength
17 FORM_WIDTH = .SCB[PSMSL_PAGE_WIDTH];
18 FORM_LENGTH = .SCB[PSMSL_PAGE_LENGTH];
19 RETURN_IF_ERROR_(ALLOCATE_PAGE(.SCB)); ! Get the page of memory
20 PAGE_REF = .SCB[PSMSA_PAGE_POINTER]; ! My local page pointer
21 ! Always start at top of page
22 PAGE_REF[0,0..FORM_WIDTH] = PSMSK_CHAR_FF; ! form feed in 0 pos.
23 ! Standard Trailer Page 132x66: text covers rows 1 through 58,
24 ! translated to frames... ref starts at 1 and length is 57.
25 FILL_JOB_TRAILER(.SCB,
26 PAGE_REF[0,1..FORM_WIDTH],
27 .FORM_WIDTH,
28 .FORM_LENGTH - 6 - 2 - 1); ! ...6 burst, 2 sp,
29 ! top margin is 1
30 END;
31 [PSMSK CLOSE]: ! Return the Page of Memory
32 RETURN_IF_ERROR_(DEALLOCATE_PAGE(.SCB));
33 [OTHERWISE]:
34 RETURN PSMS_FUNNOTSUP;
35 TES; ! case .function
36 SSS_NORMAL
37 END;

SEPARATE
V04-001Print Symbiant -- separation routines
JOB_TRAILERE 15
16-Sep-1984 02:23:03
14-Sep-1984 22:32:26VAX-11 Bliss-32 V4.0-742
[PRTSMB.SRC]SEPARATE.B32;2Page 27
(9)

							.ENTRY	PSMS\$JOB TRAILER, Save R2,R3,R4,R5	: 1784
							MOVL	@SMB_CONTEXT SCB	: 1804
							MOVL	@FUNCTION, R0	: 1808
							CMPL	R0, #5	: 1810
							BNEQ	2\$: 1814
							MOVL	508(SCB), PAGE_REF	: 1816
							SUBL3	#8, 504(SCB), R0	: 1818
							CMPL	620(SCB), R0	: 1820
							BLEQ	1\$: 1822
							MOVL	#PSMS_EOF, R0	: 1826
							RET		: 1828
							MOVL	FUNC_DESC, R3	: 1830
							MOVL	512(SCB), (R3)	: 1832
							MULL3	512(SCB), 620(SCB), R0	: 1834
							ADDL3	PAGE_REF, R0, 4(R3)	: 1836
							PUSHL	(R3)	: 1838
							PUSHL	#32	: 1840
							PUSHL	4(R3)	: 1842
							CALLS	#3, DELIMIT_STRING_NOT	: 1844
							MOVL	R0, (R3)	: 1846
							BRB	5\$: 1848
							CMPL	R0, #4	: 1850
							BNEQ	3\$: 1852
							PUSHL	SCB	: 1854
							CALLS	#1, GET_FORM_SIZE	: 1856
							MOVL	512(SCB), FORM_WIDTH	: 1858
							MOVL	504(SCB), FORM_LENGTH	: 1860
							PUSHL	SCB	: 1862
							CALLS	#1, ALLOCATE_PAGE	: 1864
							BLBC	STATUS, 6\$: 1866
							MOVL	508(SCB), PAGE_REF	: 1868
							MOVBL	#12, (PAGE_REF)	: 1870
							PUSHAB	-9(FORM_LENGTH)	: 1872
							PUSHL	FORM_WIDTH	: 1874
							PUSHAB	(FORM_WIDTH)[PAGE_REF]	: 1876
							PUSHL	SCB	: 1878
							CALLS	#4, FILL_JOB_TRAILER	: 1880
							BRB	5\$: 1882
							CMPL	R0, #2	: 1884
							BNEQ	4\$: 1886
							PUSHL	SCB	: 1888
							CALLS	#1, DEALLOCATE_PAGE	: 1890
							BLBS	STATUS, 5\$: 1892
							RET		: 1894
							MOVL	#PSMS_FUNNOTSUP, R0	: 1896
							RET		: 1898
							MOVL	#1, R0	: 1900
							RET		: 1902

; Routine Size: 165 bytes. Routine Base: CODE + 04C8

928 1866 1 %sbttl 'PSMSPAGE_HEADER - Print a Header at the Top of each Page'
929 1867 1 Functional Description:
930 1868 1 Creates a page header for the current file and prints it at the
931 1869 1 top of each page.
932 1870 1 FUNCTION:
933 1871 1 OPEN - Allocate and create the Page Header
934 1872 1 READ - Return the current header with the new page number
935 1873 1 CLOSE - Deallocate the header
936 1874 1
937 1875 1 Formal Parameters:
938 1876 1 SMB_CONTEXT - Pointer to the SMB
939 1877 1 USER_CONTEXT - User defined pointer (not used here)
940 1878 1 FUNCTION - OPEN, READ, CLOSE
941 1879 1 FUNC_DESC - Pointer to functionally dependent descriptor
942 1880 1 FUNC_ARG - Pointer to functionally dependent argument
943 1881 1
944 1882 1 Implicit Inputs:
945 1883 1 none
946 1884 1
947 1885 1 Implicit Outputs:
948 1886 1 none
949 1887 1
950 1888 1 Returned Value:
951 1889 1 none
952 1890 1
953 1891 1 Side Effects:
954 1892 1 none
955 1893 1 --
956 1894 1 GLOBAL ROUTINE PSMSPAGE_HEADER (%SBTTL 'PAGE_HEADER'
957 1895 1 SMB_CONTEXT : REF VECTOR,
958 1896 1 USER_CONTEXT : REF VECTOR,
959 1897 1 FUNCTION : REF VECTOR,
960 1898 1 FUNC_DESC : REF VECTOR,
961 1899 1 FUNC_ARG : REF VECTOR
962 1900 1) =
963 1901 2 BEGIN
964 1902 2 LOCAL
965 1903 2 SCB : REF \$BBLOCK;
966 1904 2
967 1905 2 SCB = .SMB_CONTEXT[0];
968 1906 2
969 1907 2 ! Check the FUNCTION requested
970 1908 2
971 1909 2 SELECTONEU .FUNCTION[0] OF
972 1910 2 SET
973 1911 2 [PSMSK READ]:
974 1912 2 BEGIN
975 1913 2
976 1914 2 IF .SCB[PSMSL_RECORD_NUMBER] GTRU 0
977 1915 2 THEN
978 1916 2 RETURN PSMS_EOF;
979 1917 2
980 1918 2 ! Use the supplied string descriptor as a temp for the page number
981 1919 2
982 1920 2 FUNC_DESC[SIZE] = 5;
983 1921 2 FUNC_DESC[ADDR] = .SCB_ADDR_(PAGE_HEADER) + .SCB_SIZE_(PAGE_HEADER) - 8;
984 1922 2

SEPARATE
V04-001

Print Symbiont -- separation routines
PAGE_HEADER

G 15
16-Sep-1984 02:23:03 VAX-11 Bliss-32 V4.0-742
14-Sep-1984 22:32:26 [PRTSMB.SRC]SEPARATE.832;2

Page 29
(10)

985 1923 3 ! Write the page number into the end of the page header buffer
986 1924 3 (note -- since the page number can decrease we always fill out
987 1925 3 the page number area with blanks to overwrite any prior data)
988 1926 3
989 P 1927 3 SFAD (
990 P 1928 3 \$DESCRIPTOR ('!5<!UL!>'), ! pad with trailing spaces
991 P 1929 3 FUNC_DESC[SIZE], ! ignore return length
992 P 1930 3 FUNC_DESC[0], ! temp output buffer desc
993 P 1931 3 SCB[PSMSL_PAGE] ! current page number
994 1932 3 ;
995 1933 3
996 1934 3 ! copy the page header descriptor size and address to
997 1935 3 the function descriptor
998 1936 3
999 1937 3 FUNC_DESC[SIZE] = .SCB_SIZE_(PAGE_HEADER);
1000 1938 3 FUNC_DESC[ADDR] = .SCB_ADDR_(PAGE_HEADER);
1001 1939 3
1002 1940 2 END;
1003 1941 2
1004 1942 2 [PSMSK_OPEN]:
1005 1943 3 BEGIN
1006 1944 3 ! set carriage control to imbedded
1007 1945 3
1008 1946 3 FUNC_ARG[0] = PSMSK_CC_INTERNAL;
1009 1947 3
1010 1948 3 ! Format everything but the page number, but only do it once per task
1011 1949 3
1012 1950 3 IF TESTBITCS (SCB[PSMSV_PAGE_HEADER_BUILT])
1013 1951 3 THEN
1014 1952 3 CREATE_PAGE_HEADER (.SCB);
1015 1953 3
1016 1954 2 END;
1017 1955 2
1018 1956 2 [PSMSK_START_TASK]:
1019 1957 2 ! Set the size of the page header equal to the page width adjusted
1020 1958 2 for margins.
1021 1959 2
1022 1960 3 BEGIN
1023 1961 3 GET_FORM_SIZE (.SCB); ! Returns the WidthxLength
1024 1962 3
1025 1963 3 ! Adjust for margins and imbedded carriage control
1026 1964 3
1027 1965 3 SCB[PSMSL_PAGE_WIDTH] = .SCB[PSMSL PAGE WIDTH]
1028 1966 3 - .SCB[PSMSL_LEFT_MARGIN] ! less leading spaces
1029 1967 3 - .SCB[PSMSL_RIGHT_MARGIN] ! less early truncation
1030 1968 3 + 3; ! plus trailing carr
1031 1969 3 ! cntrl <CR><LF><LF>
1032 1970 3
1033 P 1971 3 RETURN_IF_ERROR_ (STR\$GET1 DX (%REF (.SCB[PSMSL_PAGE_WIDTH]),
1034 1972 3 SCB[PSMSL_PAGE_HEADER]));
1035 1973 2 END;
1036 1974 2
1037 1975 2 [OTHERWISE]:
1038 1976 2 RETURN PSMS_FUNNOTSUP;
1039 1977 2
1040 1978 2 TES; ! case .function
1041 1979 2

**SEPARATE
V04-001**

Print Symbiont -- separation routines
PAGE_HEADER

H 15
16-Sep-1984 02:23:03 VAX-11 Bliss-32 V4.0-742
14-Sep-1984 22:32:26 [PRTSMB.SRC]SEPARATE.B32;2

Page 30
(10)

1042 1980 2 SSS_NORMAL
1043 1981 2
1044 1982 1 END;

3E	21	4C	55	21	3C	35	21	0056D	P.AAB:	.ASCII	\!5<!UL!>\	;
								00575	.BLKB	3		
								00000008	P.AAA:	.LONG	8	
								00000000	.ADDRESS P.AAB			
								0057C	.EXTRN	SYSSFAO		
5E								001C	00000	.ENTRY	PSMSPAGE_HEADER, Save R2,R3,R4	1894
52								04	C2 00002	SUBL2	#4, SP	
50								0C	BC D0 00005	MOVL	@SMB_CONTEXT SCB	1905
05								50	D1 00009	MOVL	@FUNCTION, R0	1909
								46	12 00010	CMPL	R0, #5	1911
								026C	C2 D5 00012	BNEQ	2\$	
								08	13 00016	TSTL	620(SCB)	1914
								0F	D0 00018	BEQL	1\$	
								04	0001F	MOVL	#PSMS_EOF, R0	1916
								10	AC D0 00020	RET		
								05	D0 00024	MOVL	FUNC DESC, R3	1920
								52	D0 00027	MOVL	#5, TR3)	
								01F0	C4 3C 0002A	MOVZWL	SCB, R4	
								01F4	C2 C0 0002F	ADDL2	496(R4), R0	1921
								F8	A0 9E 00034	MOVAB	500(SCB), R0	
								01EC	C2 DD 00039	PUSHL	-8(R0), 4(R3)	
									53 DD 0003D	PUSHL	492(SCB)	
									53 DD 0003F	PUSHL	R3	
								B4	AF 9F 00041	PUSHAB	R3	
								00	04 FB 00044	CALLS	P.AAA	
								63	01FB 00048	MOVZWL	#4, SYSSFAO	
								04 A3	01F0 C2 D0 00050	MOVL	496(R4), (R3)	1937
									55 11 00056	BRB	500(SCB), 4(R3)	1938
								04	50 D1 00058	CMPL	5\$	1909
									12 12 0005B	BNEQ	R0, #4	
								14 BC	01 D0 0005D	MOVL	3\$	1942
								10 A2	08 E2 00061	BBSS	#1, @FUNC ARG	
									52 DD 00066	PUSHL	#8, 16(SCB), 5\$	1946
								0000V CF	01 FB 00068	CALLS	SCB	1950
									3E 11 0006D	BRB	#1, CREATE_PAGE_HEADER	1952
								10	50 D1 0006F	CMPL	5\$	1909
									31 12 00072	BNEQ	R0, #16	
								0000V CF	52 DD 00074	PUSHL	4\$	1956
									01 FB 00076	CALLS	SCB	
								51 0200	C2 9E 0007B	MOVAB	#1, GET_FORM_SIZE	1961
								61 00BC	C2 C3 00080	SUBL3	512(SCB), R1	
								50 0148	C2 C2 00086	SUBL2	188(SCB), (R1), R0	1965
								61 03	A0 9E 00088	MOVAB	328(SCB), R0	1966
								01F0	C2 9F 0008F	PUSHAB	3(R0), (R1)	1967
									61 D0 00093	MOVL	496(SCB)	1968
								04 AE	AE 9F 00097	PUSHAB	(R1), 4(SP)	
								00 09	02 FB 0009A	CALLS	4(SP)	1972
									50 E8 000A1	BLBS	#2, STRGET1_DX	
											STATUS, 5\$	

SEPARATE
V04-001

Print Symbiont -- separation routines
PAGE_HEADER

I 15
16-Sep-1984 02:23:03 VAX-11 Bliss-32 V4.0-742
14-Sep-1984 22:32:26 [PRTSMB.SRC]SEPARATE.B32;2

Page 31
(10)

50 0000000G	8F	04 000A4	RET	
		00 000AS	MOVL	#PSMS_FUNNOTSUP, R0
		04 000AC	RET	
50	01	00 000AD	MOVL	#1, R0
		04 000B0	RET	

; Routine Size: 177 bytes, Routine Base: CODE + 0580

SEPARATE
V04-001

Print Symbiont -- separation routines
PARSE_FILE_NAME

J 15

16-Sep-1984 02:23:03
14-Sep-1984 22:32:26

VAX-11 Bliss-32 V4.0-742
[PRTSMB.SRC]SEPARATE.B32;2

Page 32
(11)

```
: 1046    1983 1 ROUTINE PARSE_FILE_NAME ( %SBTTL 'PARSE_FILE_NAME'  
.: 1047    1984 1 FILENAME : REF $BBLOCK,  
.: 1048    1985 1 ITEM_CODE  
.: 1049    1986 1 RESULT : REF VECTOR  
.: 1050    1987 1 ) =  
.: 1051    1988 2 BEGIN  
.: 1052    1989 2 LOCAL  
.: 1053    1990 2 LIST : $ITMBLK [1,8]  
.: 1054    1991 2 ;  
.: 1055    1992 2  
.: 1056    1993 2 CH$FILL (0, %ALLOCATION (LIST), LIST);  
.: 1057    1994 2  
.: 1058    1995 2 LIST [0, ITMSW_ITMCOD] = .ITEM_CODE;  
.: 1059    1996 2  
.: 1060    1997 2 RETURN_IF_ERROR_ ($FILESCAN (SRCSTR=.FILENAME, VALUELIST=LIST));  
.: 1061    1998 2  
.: 1062    1999 2  
.: 1063    2000 2 RESULT[SIZE] = .LIST[0, ITMSW_BUFSIZ];  
.: 1064    2001 2 RESULT[ADDR] = .LIST[0, ITMSL_BUFADR];  
.: 1065    2002 2  
.: 1066    2003 2 SSS_NORMAL  
.: 1067    2004 2  
.: 1068    2005 1 END;
```

.EXTRN SYSSFILESCAN

003C 00000 PARSE_FILE_NAME:

OC	00	5E	00	C2	00002	WORD	Save R2,R3,R4,R5	: 1983
		6E	00	2C	00005	SUBL2	#12, SP	: 1994
			6E		0000A	MOVCS	#0, (SP), #0, #12, LIST	
	02	AE	08	AC	80 0000B	MOVW	ITEM_CODE, LIST+2	: 1996
			04	7E	D4 00010	CLRL	-(SPT)	: 1998
			04	AE	9F 00012	PUSHAB	LIST	
	00000000G	00	04	AC	DD 00015	PUSHL	FILENAME	
		OF	03	03	FB 00018	CALLS	#3, SYSSFILESCAN	
		50	0C	50	E9 0001F	BLBC	STATUS, 1\$	
		60	0C	AC	D0 00022	MOVL	RESULT, R0	: 2000
	04	A0	04	6E	3C 00026	MOVZWL	LIST, (R0)	: 2001
		50	01	00	00029	MOVL	LIST+4, 4(R0)	: 2005
			04	00	0002E	MOVL	#1, R0	
			04	00031	1\$: RET			

; Routine Size: 50 bytes, Routine Base: CODE + 0631

SEPARATE
V04-001Print Symbiont -- separation routines
ALLOCATE_PAGE - Allocate the Page of MemoryK 15
16-Sep-1984 02:23:03
14-Sep-1984 22:32:26VAX-11 Bliss-32 V4.0-742
[PRTSMB.SRC]SEPARATE.B32;2Page 33
(12)

```

1070 2006 1 %sbttl 'ALLOCATE_PAGE - Allocate the Page of Memory'
1071 2007 1 ++
1072 2008 1 Functional Description:
1073 2009 1 This routine allocates memory in an amount of
1074 2010 1 memory equal to the largest form Size supported.
1075 2011 1
1076 2012 1 Formal Parameters:
1077 2013 1 SCB - Address of the SCB
1078 2014 1
1079 2015 1 Implicit Inputs:
1080 2016 1 none
1081 2017 1
1082 2018 1 Implicit Outputs:
1083 2019 1 none
1084 2020 1
1085 2021 1 Returned Value:
1086 2022 1 none
1087 2023 1
1088 2024 1 Side Effects:
1089 2025 1 none
1090 2026 1 --
1091 2027 1 ROUTINE ALLOCATE PAGE(
1092 2028 1 SCB : REF $BBLOCK
1093 2029 1 ) =
1094 2030 2 BEGIN
1095 2031 2 LOCAL
1096 2032 2 PAGE_SIZE;
1097 2033 2
1098 2034 2 PAGE_SIZE = .SCB[PSMSL_PAGE_WIDTH] + (.SCB[PSMSL_PAGE_LENGTH]+1);
1099 2035 2
P 2036 2 RETURN_IF_ERROR_( LIB$GET_VM( XREF(.PAGE_SIZE),
1100 2037 2 SCB[PSMSA_PAGE_POINTER]); ! Fill it with Blanks
1101 2038 2
1102 2039 2 CHSFILL (%CHAR(32), .PAGE_SIZE,
1103 2040 2 .SCB[PSMSA_PAGE_POINTER]);
1104 2041 2
1105 2042 2 RETURN SSS_NORMAL;
1106 2043 1 END;
1107

```

003C 00000 ALLOCATE_PAGE:

						WORD	Save R2,R3,R4,R5	: 2027
						SUBL2	#4, SP	: 2034
						MOVL	SCB, R3	: 2037
						ADDL3	#1, 504(R3), R0	
						MULL3	512(R3), R0, PAGE_SIZE	
						PUSHAB	508(R3)	
						MOVL	PAGE_SIZE, 4(SP)	
						PUSHAB	4(SP)	
						CALLS	#2, LIB\$GET_VM	
						BLBC	STATUS, 1\$	
						MOVCS	#0, (SP), #32, PAGE_SIZE, a508(R3)	
						MOVL	#1, R0	

SEPARATE
V04-001

Print Symbiont -- separation routines
ALLOCATE_PAGE - Allocate the Page of Memory

L 15
16-Sep-1984 02:23:03
14-Sep-1984 22:32:26

VAX-11 Bliss-32 V4.0-742
[PRTSMB.SRC]SEPARATE.B32;2

Page 34
(12)

; 2043

04 00035 1\$: RET

; Routine Size: 54 bytes, Routine Base: CODE + 0663

S
V

;

SEPARATE
VO4-001

Print Symbiont -- separation routines
CREATE_PAGE_HEADER - Allocate and Format the Pa

N 15

16-Sep-1984 02:23:03
14-Sep-1984 22:32:26

VAX-11 Bliss-32 V4.0-742
[PRTSMB.SH]SEPARATE.B32;2

Page 36
(14)

```
1145 2079 1 %sbttl 'CREATE_PAGE_HEADER - Allocate and Format the Page Header'
1146 2080 1 ++
1147 2081 1 Functional Description:
1148 2082 1 This routine allocates memory and formats the information
1149 2083 1 for the page header. Returns success if allocation of memory
1150 2084 1 was successful.
1151 2085 1
1152 2086 1 Formal Parameters:
1153 2087 1 SCB - Address of the SCB
1154 2088 1
1155 2089 1 Implicit Inputs:
1156 2090 1 none
1157 2091 1
1158 2092 1 Implicit Outputs:
1159 2093 1 none
1160 2094 1
1161 2095 1 Returned Value:
1162 2096 1 none
1163 2097 1
1164 2098 1 Side Effects:
1165 2099 1 none
1166 2100 1 --
1167 2101 1 ROUTINE CREATE_PAGE_HEADER (
1168 2102 1 SCB : REF $BBLOCK
1169 2103 1 ) =
1170 2104 2 BEGIN
1171 2105 2
1172 2106 2 LOCAL
1173 2107 2 REMAINING,
1174 2108 2 NAME_LENGTH : ! Remaining header space
1175 2109 2 BUFFER : VECTOR [512,byte], ! Trimmed file name length
1176 2110 2 STR_DESC : VECTOR [2], ! Assume max size 512 bytes0
1177 2111 2 HEADER_REF : REF PAGE_ARRAY,
1178 2112 2 HEADER_SIZE :
1179 2113 2
1180 2114 2 !*! SMALL WIDTHS -- THE PAGE NUMBER SHOULD BE THE ONLY THING PRINTED
1181 2115 2 !*! WHEN THE WIDTH IS TOO SMALL. DATE vs. FILENAME IS DEVO'S CHOICE
1182 2116 2
1183 2117 2 HEADER_SIZE = .SCB_SIZE_(PAGE_HEADER) - 3; ! don't include the carriage
1184 2118 2 control area of 3 bytes
1185 2119 2 HEADER_REF = .SCB_ADDR_(PAGE_HEADER);
1186 2120 2
1187 2121 2 CH$FILL (%CHAR(32), .HEADER_SIZE, .HEADER_REF);
1188 2122 2
1189 2123 2 ! Insert imbedded carriage control <LF><LF><CR>
1190 2124 2
1191 2125 2 CH$FILL (PSMSK_CHAR_LF, 2, (.HEADER_REF + .HEADER_SIZE));
1192 2126 2 CH$FILL (PSMSK_CHAR_CR, 1, (.HEADER_REF + .HEADER_SIZE) + 2);
1193 2127 2 ! address is offset by two
1194 2128 2
1195 2129 2 ! If the header is too small even for "Page 99999" then disable page
1196 2130 2 headers. (Maybe this code should be in "MESSAGE").
1197 2131 2
1198 2132 2 IF .HEADER_SIZE LSSU 10
1199 2133 2 THEN
1200 2134 2 RETURN SSS_NORMAL;
1201 2135 2
```

```
: 1202 2136 2 ! Set up the buffer descriptor for "GET_xxx" Routines
: 1203 2137 2
: 1204 2138 2 STR_DESC[SIZE] = XALLOCATION(BUFFER);           ! allocate for routines
: 1205 2139 2 STR_DESC[ADDR] = BUFFER;                      ! init address
: 1206 2140 2
: 1207 2141 2 ! Insert the word "Page"
: 1208 2142 2
: 1209 2143 2 MOVE_FRAME
: 1210 2144 2     (.SCB,
: 1211 2145 2     $DESCRIPTOR ('Page '),
: 1212 2146 2     HEADER_REF[HEADER_SIZE-10,0..SCB[PSMSL_FORM_WIDTH]]).
: 1213 2147 2     5
: 1214 2148 2     .1$;
: 1215 2149 2
: 1216 2150 2 ! Get the filename - include the expected length
: 1217 2151 2
: 1218 2152 2 STR_DESC[SIZE] = XALLOCATION(BUFFER);           ! reset buffer length
: 1219 2153 2 GET_FILE_NAME
: 1220 2154 2     T(SCB,
: 1221 2155 2     .HEADER_SIZE - 11,                         ! SCB addr.
: 1222 2156 2     STR_DESC[0],                           ! expected length (less page #)
: 1223 2157 2     STR_DESC[SIZE]);                     ! Buffer descriptor
: 1224 2158 2                                         ! Returned length
: 1225 2159 2 NAME_LENGTH = .STR_DESC[SIZE] + 1;          ! Save the trimmed length
: 1226 2160 2
: 1227 2161 2 INSERT_FRAME
: 1228 2162 2     (.SCB,
: 1229 2163 2     STR_DESC[0],
: 1230 2164 2     HEADER_REF[0,0..SCB[PSMSL_FORM_WIDTH]],
: 1231 2165 2     .NAME_LENGTH - 1,                         ! Always less than frame_width
: 1232 2166 2     1);
: 1233 2167 2
: 1234 2168 2
: 1235 2169 2 ! The area remaining for the date is the original header
: 1236 2170 2 width less the size of the file name, less the size for the page
: 1237 2171 2 number field ('Page 99999') less one blank for each.
: 1238 2172 2
: 1239 2173 2 REMAINING = .HEADER_SIZE - .NAME_LENGTH - 10 - 1;
: 1240 2174 2 IF .REMAINING LSSU T8
: 1241 2175 2 THEN
: 1242 2176 2     RETURN SSS_NORMAL;
: 1243 2177 2
: 1244 2178 2 ! Get the file revision date and center it between file name and page number
: 1245 2179 2
: 1246 2180 2 STR_DESC[SIZE] = XALLOCATION(BUFFER);           ! reset buffer length
: 1247 2181 2 GET_REVISION_DATE
: 1248 2182 2     (.SCB,
: 1249 2183 2     STR_DESC[0],                           ! SCB addr.
: 1250 2184 2     STR_DESC[SIZE]);                     ! Buffer descriptor
: 1251 2185 2                                         ! Returned length
: 1252 2186 2 CENTER_FRAME
: 1253 2187 2     (.SCB,
: 1254 2188 2     STR_DESC[0],
: 1255 2189 2     HEADER_REF[.NAME_LENGTH,0..SCB[PSMSL_FORM_WIDTH]],
: 1256 2190 2     .REMAINING,
: 1257 2191 2     1);
: 1258 2192 2
```

```
1259 : 2193 2 RETURN SSS_NORMAL;
1260 : 2194 2
1261 : 2195 1 END;
```

20	65	67	61	50	006C6	P.AAD:	.ASCII \Page \
					006CB	.BLKB	1
					00000005	P.AAC:	.LONG 5
					00000000	.006D0	.ADDRESS P.AAD

01FC 00000 CREATE_PAGE HEADER:								
								2101
								2117
								2119
								2121
56	20	6E	5E	FDF8	CE 9E 00002	.WORD	Save R2,R3,R4,R5,R6,R7,R8	
			58	04	AC D0 00007	MOVAB	-520(SP), SP	
			50	01F0	C8 9E 0000B	MOVL	SCB, R8	
			56		60 3C 00010	MOVAB	496(R8), R0	
			56		03 C2 00013	MOVZWL	(R0), HEADER_SIZE	
			57	04	A0 D0 00016	SUBL2	#3, HEADER_SIZE	
					00 2C 0001A	MOVL	4(R0), HEADER_REF	
					67 0001F	MOVCS	#0, (SP), #32, HEADER_SIZE, (HEADER_REF)	
						PUSHAB	(HEADER_SIZE)[HEADER_REF]	2125
			50	9E	0AOA 6647 9F 00020	MOVW	#2570, 8(SP)+	
				57	8F B0 00023	ADDL3	HEADER_SIZE, HEADER_REF, R0	
				A0	56 C1 00028	MOVB	#13, 27R0)	2126
				0A	0D 90 0002C	CMPL	HEADER_SIZE, #10	2132
					56 D1 00030	BLSSU	1\$	
					72 1F 00033	MOVZWL	#512, STR_DESC	2138
			04	6E	0200 8F 3C 00035	MOVAB	BUFFER, STR_DESC+4	2139
				AE	08 AE 9E 0003A	PUSHL	#1	2146
					01 DD 0003F	PUSHL	#5	
					05 DD 00041	PUSHAB	-10(HEADER_SIZE)[HEADER_REF]	
					F6 A647 9F 00043	PUSHAB	P.AAC	2145
					AE AF 9F 00047	PUSHL	R8	2146
			0000V	CF	58 DD 0004A	CALLS	#5, MOVE_FRAME	
				6E	0200 05 FB 0004C	MOVZWL	#512, STR_DESC	2152
					8F 3C 00051	PUSHL	SP	2157
					5E DD 00056	PUSHAB	STR_DESC	2156
				04	AE 9F 00058	PUSHAB	-11(HEADER_SIZE)	2155
				F5	A6 9F 0005B	PUSHL	R8	2154
					58 DD 0005E	CALLS	#4, GET_FILE_NAME	
			0000V	CF	04 FB 00060	ADDL3	#1, STR_DESC, NAME_LENGTH	2159
				6E	01 C1 00065	PUSHL	#1	2164
					C1 DD 00069	PUSHAB	-1(NAME LENGTH)	2165
					FF A2 9F 0006B	PUSHL	HEADER_REF	2164
					57 DD 0006E	PUSHAB	STR_DESC	2163
				OC	AE 9F 00070	PUSHL	R8	2164
			0000V	CF	58 DD 00073	CALLS	#5, INSERT_FRAME	
					05 FB 00075	SUBL2	NAME_LENGTH, R6	2173
					52 C2 0007A	SUBL2	#11, REMAINING	
					08 C2 0007D	CMPL	REMAINING, #18	2174
					56 D1 00080	BLSSU	1\$	
					22 1F 00083	MOVZWL	#512, STR_DESC	2180
				6E	0200 8F 3C 00085	PUSHL	SP	2184
					5E DD 0008A	PUSHAB	STR_DESC	2183
					04 AE 9F 0008C			

SEPARATE
V04-001

Print Symbiont -- separation routines
CREATE_PAGE_HEADER - Allocate and Format the Pa

D 16

16-Sep-1984 02:23:03
14-Sep-1984 22:32:26

VAX-11 Bliss-32 v4.0-742
[PRTSMB.SRC]SEPARATE.B32:2

Page 39
(14)

0000V CF	58 DD 0008F	PUSHL R8	
	03 FB 00091	CALLS #3, GET_REVISION_DATE	: 2182
	01 DD 00096	PUSHL #1	: 2189
	56 DD 00098	PUSHL REMAINING	: 2190
	CC 6247 9F 0009A	PUSHAB (NAME LENGTH)[HEADER_REF]	: 2189
	AE 9F 0009D	PUSHAB STR_DESC	: 2188
0000V CF	58 DD 000A0	PUSHL R8	: 2189
	05 FB 000A2	CALLS #5, CENTER_FRAME	
	01 D0 000A7 1\$:	MOVL #1, R0	: 2193
	04 000AA	RET	: 2195

: Routine Size: 171 bytes, Routine Base: CODE + 06D4

```
1263 2196 1 %sbttl 'FILL_FILE_FLAG - Insert Information into the FILE Page'
1264 2197 1 ++
1265 2198 1 Functional Description:
1266 2199 1 This procedure controls all inserts required for the FILE Page.
1267 2200 1
1268 2201 1 Formal Parameters:
1269 2202 1 SCB - Address of the SCB
1270 2203 1 PAGE_REF - Pointer to the Page (first byte)
1271 2204 1 PAGE_LENGTH - Length of Frame
1272 2205 1 PAGE_WIDTH - Width of Frame
1273 2206 1
1274 2207 1 Implicit Inputs:
1275 2208 1 none
1276 2209 1
1277 2210 1 Implicit Outputs:
1278 2211 1 none
1279 2212 1
1280 2213 1 Returned Value:
1281 2214 1 none
1282 2215 1
1283 2216 1 Side Effects:
1284 2217 1 none
1285 2218 1 --
1286 2219 1 ROUTINE FILL_FILE_FLAG (
1287 2220 1 SCB : REF $BBLOCK,
1288 2221 1 PAGE_REF : REF PAGE_ARRAY,
1289 2222 1 PAGE_WIDTH,
1290 2223 1 PAGE_LENGTH
1291 2224 1 ): NOVALUE =
1292 2225 2 BEGIN
1293 2226 2 LITERAL K_MAX_BUFFER_SIZE = 512;
1294 2228 2
1295 2229 2 LOCAL
1296 2230 2 RET_LEN : VECTOR[1],
1297 2231 2 TOP_OFFSET,
1298 2232 2 BOTTOM_OFFSET,
1299 2233 2 BUFFER : VECTOR [512,byte], ! Assume max size 512 bytes
1300 2234 2 STRING_DESC : VECTOR [2]; ! Descriptor to current string
1301 2235 2
1302 2236 2 ! Allocate the buffer for "GET_xxx" Routines
1303 2237 2
1304 2238 2 STRING_DESC[SIZE] = XALLOCATION(BUFFER); ! allocate for routines
1305 2239 2 STRING_DESC[ADDR] = BUFFER; ! init address
1306 2240 2
1307 2241 2 TOP_OFFSET = 0; ! start insert at zero
1308 2242 2 BOTTOM_OFFSET = .PAGE_LENGTH - 2; ! Note: offset includes next
1309 2243 2 ! "insert" frame length
1310 2244 2
1311 2245 2 ! Burst characters
1312 2246 2
1313 2247 2 FILL_FRAME (.SCB,
1314 2248 2 SCB[PSMSB_FILE_BURST_CHAR],
1315 2249 2 PAGE_REF[0,.TOP_OFFSET,.PAGE_WIDTH], .PAGE_WIDTH, 3);
1316 2250 2
1317 2251 2 FILL_FRAME (.SCB,
1318 2252 2 SCB[PSMSB_FILE_BURST_CHAR],
```

SEPARATE
V04-001

F 16
16-Sep-1984 02:23:03 VAX-11 Bliss-32 V4.0-742
FILL_FILE_FLAG - Insert Information into the FI 14-Sep-1984 22:32:26 [PRTSMB.SRC]SEPARATE.B32:2

```
1320      2253 2          PAGE_REF[0,.BOTTOM_OFFSET,.PAGE_WIDTH], .PAGE_WIDTH, 3);  
1321      2254 2  
1322      2255 2          FILL_FRAME (.SCB,  
1323      2256 2              PAGE_REF[10,.TOP_OFFSET,.PAGE_WIDTH], .PAGE_WIDTH-20, 3);  
1324      2257 2  
1325      2258 2          FILL_FRAME (.SCB,  
1326      2259 2              PAGE_REF[10,.BOTTOM_OFFSET,.PAGE_WIDTH], .PAGE_WIDTH-20, 3);  
1327      2260 2  
1328      2261 2          FILL_FRAME (.SCB,  
1329      2262 2              PAGE_REF[10,.TOP_OFFSET,.PAGE_WIDTH], .PAGE_WIDTH-28, 3);  
1330      2263 2  
1331      2264 2          FILL_FRAME (.SCB,  
1332      2265 2              .SCB[PSMSB_JOB_BURST_CHAR]  
1333      2266 2                  PAGE_REF[14,.TOP_OFFSET,.PAGE_WIDTH], .PAGE_WIDTH-28, 3);  
1334      2267 2  
1335      2268 2          FILL_FRAME (.SCB,  
1336      2269 2              .SCB[PSMSB_JOB_BURST_CHAR],  
1337      2270 2                  PAGE_REF[14,.BOTTOM_OFFSET,.PAGE_WIDTH], .PAGE_WIDTH-28, 3);  
1338      2271 2          ! Get the sys$announce note and output to page  
1339      2272 2          note: system announcement will fit or will be truncated so there is  
1340      2273 2          no updating of "offsets"  
1341      2274 2  
1342      2275 2          ! re-init  
1343      2276 2          STRING_DESC[SIZE] = K_MAX_BUFFER_SIZE; ! reset buffer size  
1344      2277 2  
1345      2278 2          GET_SYSTEM_ANNOUNCEMENT  
1346      2279 2          (.SCB,  
1347      2280 2              STRING_DESC[0],           ! SCB addr.  
1348      2281 2              STRING_DESC[SIZE]);        ! Buffer descriptor  
1349      2282 2          STRING_DESC[SIZE];       ! Returned length  
1350      2283 2          CENTER_FRAME (.SCB,  
1351      2284 2              STRING_DESC[0],  
1352      2285 2                  PAGE_REF[0,.TOP_OFFSET + 1,.PAGE_WIDTH], .PAGE_WIDTH, 1);  
1353      2286 2  
1354      2287 2          ! re-init  
1355      2288 2          STRING_DESC[SIZE] = K_MAX_BUFFER_SIZE; ! reset buffer size  
1356      2289 2  
1357      2290 2          GET_DIGITAL_LOGO  
1358      2291 2          (.SCB,  
1359      2292 2              STRING_DESC[0],           ! SCB addr.  
1360      2293 2              STRING_DESC[SIZE]);        ! Buffer descriptor  
1361      2294 2          STRING_DESC[SIZE];       ! Returned length  
1362      2295 2          CENTER_FRAME (.SCB,  
1363      2296 2              STRING_DESC[0],  
1364      2297 2                  PAGE_REF[0,.BOTTOM_OFFSET + 1,.PAGE_WIDTH], .PAGE_WIDTH, 1);  
1365      2298 2  
1366      2299 2  
1367      2300 2          ! Create a sentence describing the current job.  
1368      2301 2  
1369      2302 2          ! re-init  
1370      2303 2          STRING_DESC[SIZE] = K_MAX_BUFFER_SIZE;           ! reset buffer size  
1371      2304 2          TOP_OFFSET = .TOP_OFFSET + 4;                      ! adjust & allow for spacing  
1372      2305 2  
1373      2306 2          GET_JOB_DESCRIPTION  
1374      2307 2          (.SCB,  
1375      2308 2              STRING_DESC[0],           ! SCB addr.  
1376      2309 2                  ! Use present tense  
1377      2310 2              STRING_DESC[SIZE]);        ! Buffer descriptor
```

Page 41
(15)

G 16
16-Sep-1984 02:23:03 VAX-11 Bliss-32 V4.0-742
FILL_FILE_FLAG - Insert Information into the FI 14-Sep-1984 22:32:26 [PRTSMB.SRC]SEPARATE.B32;2

```
1377      2310 2           STRING_DESC[SIZE]); ! Returned length
1378      2311 2
1379      2312 2           RET_LEN[0] = RETURN_FRAME_LENGTH
1380      2313 2           (.SCB,
1381      2314 2           STRING_DESC[0],
1382      2315 2           PAGE_REF[0,0..PAGE_WIDTH],    | string ref.
1383      2316 2           .PAGE_WIDTH,                  | ref to frame
1384      2317 2           .BOTTOM_OFFSET - .TOP_OFFSET); | cols to fill
1385      2318 2           .RET_LEN[0];                   | rows to fill
1386      2319 2           IF .RET_LEN[0] GTR 0
1387      2320 2           THEN
1388      2321 2           BOTTOM_OFFSET = .BOTTOM_OFFSET - (.RET_LEN[0] + 1);
1389      2322 2           ! offset before inserting
1390      2323 2           ! includes the space
1391      2324 2           ! Insert the string delimited. Bottom of page.
1392      2325 2           INSERT_FRAME (.SCB,
1393      2326 2           STRING_DESC[0],
1394      2327 2           PAGE_REF[0..BOTTOM_OFFSET..PAGE_WIDTH], | string ref.
1395      2328 2           .PAGE_WIDTH,                  | ref to frame
1396      2329 2           .RET_LEN[0]);                   | cols to fill
1397      2330 2           .RET_LEN[0];                   | rows to fill
1398      2331 2
1399      2332 2           ! Create a sentence describing the current file. Bottom of page.
1400      2333 2
1401      2334 2           ! re-init
1402      2335 2           STRING_DESC[SIZE] = K_MAX_BUFFER_SIZE; ! reset buffer size
1403      2336 2
1404      2337 2           GET_FILE_DESCRIPTION
1405      2338 2           (.SCB,
1406      2339 2           STRING_DESC[0],
1407      2340 2           STRING_DESC[SIZE]);          | SCB addr.
1408      2341 2           ! Buffer descriptor
1409      2342 2           RET_LEN[0] = RETURN_FRAME_LENGTH
1410      2343 2           (.SCB,
1411      2344 2           STRING_DESC[0],
1412      2345 2           PAGE_REF[0,0..PAGE_WIDTH],    | string ref.
1413      2346 2           .PAGE_WIDTH,                  | ref to frame
1414      2347 2           .BOTTOM_OFFSET - .TOP_OFFSET); | cols to fill
1415      2348 2           .RET_LEN[0];                   | rows to fill
1416      2349 2           IF .RET_LEN[0] GTR 0
1417      2350 2           THEN
1418      2351 2           BOTTOM_OFFSET = .BOTTOM_OFFSET - (.RET_LEN[0] + 1);
1419      2352 2           ! offset before inserting
1420      2353 2
1421      2354 2           ! insert the string delimited
1422      2355 2           INSERT_FRAME (.SCB,
1423      2356 2           STRING_DESC[0],
1424      2357 2           PAGE_REF[0..BOTTOM_OFFSET..PAGE_WIDTH], | string ref.
1425      2358 2           .PAGE_WIDTH,                  | ref to frame
1426      2359 2           .RET_LEN[0]);                   | cols to fill
1427      2360 2           .RET_LEN[0];                   | rows to fill
1428      2361 2
1429      2362 2           ! User note
1430      2363 2
1431      2364 2           ! re-init
1432      2365 2           STRING_DESC[SIZE] = K_MAX_BUFFER_SIZE; ! reset buffer size
1433      2366 2
```

```
1434 2367 2 ! Get the user note
1435 2368 2 GET_USER_NOTE
1436 2369 2 (.SCB,
1437 2370 2 STRING_DESC[0],
1438 2371 2 STRING_DESC[SIZE]); | SCB addr.
1439 2372 2 | Buffer descriptor
1440 2373 2 | Returned length
1441 2374 2 RET_LEN[0] = RETURN_FRAME_LENGTH
1442 2375 2 (.SCB,
1443 2376 2 STRING_DESC[0],
1444 2377 2 PAGE_REF[0,0,.PAGE_WIDTH], | string ref.
1445 2378 2 .PAGE_WIDTH, | ref to frame
1446 2379 2 .BOTTOM_OFFSET -.TOP_OFFSET); | cols to fill
1447 2380 2 ! insert the string delimited
1448 2381 2 INSERT_FRAME (.SCB,
1449 2382 2 STRING_DESC[0], | string ref.
1450 2383 2 PAGE_REF[0,.TOP_OFFSET,.PAGE_WIDTH], | ref to frame
1451 2384 2 .PAGE_WIDTH, | cols to fill
1452 2385 2 .RET_[EN[0]]; | rows to fill
1453 2386 2
1454 2387 2
1455 2388 2 ! User name
1456 2389 2
1457 2390 2 ! re-init
1458 2391 2 STRING_DESC[SIZE] = K_MAX_BUFFER_SIZE; | reset buffer size
1459 2392 2 IF .RET_LEN[0] GTR 0
1460 2393 2 THEN
1461 2394 2 TOP_OFFSET = .TOP_OFFSET + .RET_LEN[0] + 1
1462 2395 2 ELSE
1463 2396 2 TOP_OFFSET = .TOP_OFFSET + 1; | adjust & allow for spacing
1464 2397 2
1465 2398 2
1466 2399 2 RET_LEN[0] = INSERT_NAME_BANNER (.SCB,
1467 2400 2 SCB_SIZE(USER NAME), | user name desc
1468 2401 2 PAGE_REF[0,.TOP_OFFSET,.PAGE_WIDTH], | ref to frame
1469 2402 2 .PAGE_WIDTH, | max width Bann
1470 2403 2 .BOTTOM_OFFSET -.TOP_OFFSET, | frame length
1471 2404 2 7); | max hght Bann str desired
1472 2405 2
1473 2406 2
1474 2407 2
1475 2408 2
1476 2409 2 IF .RET_LEN[0] GTR 0
1477 2410 2 THEN
1478 2411 2 TOP_OFFSET = .TOP_OFFSET + .RET_LEN[0] + 2; | adjust & allow for spacing
1479 2412 2
1480 2413 2 ! Get and insert the filename banner
1481 2414 2
1482 2415 2 ! re-init
1483 2416 2 STRING_DESC[SIZE] = K_MAX_BUFFER_SIZE; | reset buffer size
1484 2417 2 RET_LEN[0] = INSERT_FILENAME_BANNER
1485 2418 2 (.SCB,
1486 2419 2 STRING_DESC[0], | Buffer desc.
1487 2420 2 PAGE_REF[0,.TOP_OFFSET,.PAGE_WIDTH], | ref to frame
1488 2421 2 .PAGE_WIDTH, | max width Bann
1489 2422 2 .BOTTOM_OFFSET -.TOP_OFFSET);
```


**SEPARATE
V04-001**

Print_Symbiont -- separation routines
FILL_FILE_FLAG - Insert Information in

J 16

16-Sep-1984 02:23:03
14-Sep-1984 22:32:26

VAX-11 Bliss-32 V4.0-742
[PRTSMB.SRC]SEPARATE.B32;2

Page 45
(15)

SEPARATE
V04-001Print Symbiont -- separation routines
FILL_FILE_FLAG - Insert Information into the FI

K 16

16-Sep-1984 02:23:03
14-Sep-1984 22:32:26VAX-11 Bliss-32 V4.0-742
[PRTSMB.SRC]SEPARATE.B32;2Page 46
(15)

			04	AE 9F 00142	PUSHAB	STRING_DESC		2339
			56	DD 00142	PUSHL	R6		2338
			03	FB 00147	CALLS	#3, GET FILE DESCRIPTION		
			52	C3 0014C	SUBL3	TOP_OFFSET, BOTTOM_OFFSET, -(SP)		2347
			8F	BB 00150	PUSHR	#^MZR4,R7>		2345
			AE	9F 00154	PUSHAB	STRING_DESC		2344
			56	DD 00157	PUSHL	R6		2345
			05	FB 00159	CALLS	#5, RETURN FRAME_LENGTH		
			50	DO 0015C	MOVL	R0, RET_LEN		
			08	15 0015F	BLEQ	28		2349
			55	C3 00161	SUBL3	RET_LEN, BOTTOM_OFFSET, R0		2351
			53	A0 9E 00165	MOVAB	-1(R0), BOTTOM_OFFSET		
			55	DD 00169	PUSHL	RET_LEN		2360
			FF		PUSHL	R7		2359
			53		MULL3	R7, BOTTOM_OFFSET, R0		2357
			53		PUSHAB	(R0)[R4]		2356
			53		PUSHAB	STRING_DESC		2357
			53		PUSHL	R6		2365
			53		CALLS	#5, INSERT FRAME		2371
			53		MOVZWL	#512, STRING_DESC		2370
			53		SP			2369
			53		PUSHAB	STRING_DESC		2378
			53		PUSHL	R6		2376
			53		CALLS	#3, GET USER NOTE		2375
			53		SUBL3	TOP_OFFSET, BOTTOM_OFFSET, -(SP)		2376
			53		PUSHR	#^MZR4,R7>		
			53		PUSHAB	STRING_DESC		
			53		PUSHL	R6		2386
			53		CALLS	#5, RETURN FRAME_LENGTH		2385
			53		MOVL	R0, RET_LEN		2383
			53		PUSHL	RET_LEN		2382
			52		PUSHL	R7		2383
			52		MULL3	R7, TOP_OFFSET, R8		
			52		PUSHAB	(R8)[R4]		
			52		PUSHAB	STRING_DESC		
			52		PUSHL	R6		2391
			52		CALLS	#5, INSERT FRAME		2392
			52		MOVZWL	#512, STRING_DESC		
			52		TSTL	RET_LEN		2394
			52		BLEQ	38		
			52		MOVAB	1(RET_LEN)[TOP_OFFSET], TOP_OFFSET		
			52		BRB	48		2396
			52		INCL	TOP_OFFSET		2401
			52		PUSHL	#7		2404
			52		SUBL3	TOP_OFFSET, BOTTOM_OFFSET, -(SP)		
			52		PUSHL	R7		2403
			52		MULL3	R7, TOP_OFFSET, R8		
			52		PUSHAB	(R8)[R4]		2401
			52		PUSHAB	364(R6)		
			52		PUSHL	R6		2400
			52		CALLS	#6, INSERT NAME_BANNER		2401
			52		MOVL	R0, RET_LEN		
			52		BLEQ	58		2408
			52		MOVAB	2(RET_LEN)[TOP_OFFSET], TOP_OFFSET		2410
			52		MOVZWL	#512, STRING_DESC		2416
			52		SUBL3	TOP_OFFSET, BOTTOM_OFFSET, -(SP)		2423
			52		PUSHL	R7		2422

SEPARATE
V04-001

Print Symbiont -- separation routines
FILL_FILE_FLAG - Insert Information into the FI

L 16

16-Sep-1984 02:23:03
14-Sep-1984 22:32:26

VAX-11 Bliss-32 V4.0-742
[PRTSMB.SRC]SEPARATE.B32;2

Page 47
(15)

58	52	57	C5 001F4	MULL3	R7, TOP_OFFSET, R8	: 2420
		6844	9F 001F8	PUSHAB	(R8)[R4]	: 2419
		0C	AE 9F 001FB	PUSHAB	STRING_DESC	: 2420
			56 DD 001FE	PUSHL	R6	
0000V	CF	05	FB 00200	CALLS	#5, INSERT_FILENAME_BANNER	
	55	50	DO 00205	MOVL	R0, RET_LEN	
		05	15 00208	BLEQ	6\$: 2425
	52	02 A542	9E 0020A	MOVAB	2(RET_LEN)[TOP_OFFSET], TOP_OFFSET	: 2427
		04 0020F	68:	RET		: 2430

: Routine Size: 528 bytes. Routine Base: CODE + 077F

```
1499    2431 1 %sbttl 'FILL_JOB_FLAG - Insert Information into the JOB Page'
1500    2432 1 ++
1501    2433 1 Functional Description:
1502    2434 1 This procedure controls all inserts required for the JOB Page.
1503    2435 1
1504    2436 1 Formal Parameters:
1505    2437 1 SCB      - Address of the SCB
1506    2438 1 PAGE_REF - Pointer to the Page (first byte)
1507    2439 1 PAGE_LENGTH - Length of Frame
1508    2440 1 PAGE_WIDTH - Width of Frame
1509    2441 1
1510    2442 1 Implicit Inputs:
1511    2443 1      none
1512    2444 1
1513    2445 1 Implicit Outputs:
1514    2446 1      none
1515    2447 1
1516    2448 1 Returned Value:
1517    2449 1      none
1518    2450 1
1519    2451 1 Side Effects:
1520    2452 1      none
1521    2453 1 !--
1522    2454 1 ROUTINE FILL_JOB_FLAG (
1523    2455 1      SCB      : REF $BLOCK,
1524    2456 1      PAGE_REF : REF PAGE_ARRAY,
1525    2457 1      PAGE_WIDTH,
1526    2458 1      PAGE_LENGTH
1527    2459 1      ): NOVALUE =
1528    2460 2 BEGIN
1529    2461 2
1530    2462 2 LITERAL K_MAX_BUFFER_SIZE = 512;
1531    2463 2
1532    2464 2 LOCAL
1533    2465 2      RET_LEN   : VECTOR[1],
1534    2466 2      TOP_OFFSET,
1535    2467 2      BOTTOM_OFFSET,
1536    2468 2      BUFFER    : VECTOR [512,byte],           ! Assume max size 512 bytes
1537    2469 2      STRING_DESC : VECTOR [2];           ! Descriptor to current string
1538    2470 2
1539    2471 2 ! Allocate the buffer for "GET_xxx" Routines
1540    2472 2
1541    2473 2 STRING_DESC[SIZE] = %ALLOCATION(BUFFER);       ! allocate for routines
1542    2474 2 STRING_DESC[ADDR] = BUFFER;                  ! init address
1543    2475 2
1544    2476 2 TOP_OFFSET = 0;
1545    2477 2 BOTTOM_OFFSET = .PAGE_LENGTH - 2;          ! offset includes burst offset
1546    2478 2
1547    2479 2 ! Burst character
1548    2480 2
1549    2481 2      FILL_FRAME (.SCB,
1550    2482 2          .SCB[PSMSB_JOB_BURST_CHAR],
1551    2483 2          PAGE_REF[0-.TOP_OFFSET,.PAGE_WIDTH], .PAGE_WIDTH, 3);
1552    2484 2
1553    2485 2      FILL_FRAME (.SCB,
1554    2486 2          .SCB[PSMSB_JOB_BURST_CHAR],
1555    2487 2          PAGE_REF[0-.BOTTOM_OFFSET,.PAGE_WIDTH], .PAGE_WIDTH, 3);
```

```
1556 2488 2 : System announcement
1557 2489 2 ! re-init
1558 2490 2 STRING_DESC[SIZE] = K_MAX_BUFFER_SIZE; ! reset buffer size
1559 2491 2 GET_SYSTEM_ANNOUNCEMENT (.SCB,
1560 2492 2 STRING_DESC[0], ! SCB addr.
1561 2493 2 STRING_DESC[SIZE]); ! Buffer descriptor
1562 2494 2 ! Returned length
1563 2495 2 CENTER_FRAME (.SCB,
1564 2496 2 STRING_DESC[0],
1565 2497 2 PAGE_REF[0,.TOP_OFFSET + 1,.PAGE_WIDTH], .PAGE_WIDTH, 1);
1566 2498 2
1567 2499 2 ! re-init
1568 2500 2 STRING_DESC[SIZE] = K_MAX_BUFFER_SIZE; ! reset buffer size
1569 2501 2 GET_DIGITAL_LOGO (.SCB,
1570 2502 2 STRING_DESC[0], ! SCB addr.
1571 2503 2 STRING_DESC[SIZE]); ! Buffer descriptor
1572 2504 2 ! Returned length
1573 2505 2 CENTER_FRAME (.SCB,
1574 2506 2 STRING_DESC[0],
1575 2507 2 PAGE_REF[0,.BOTTOM_OFFSET + 1,.PAGE_WIDTH], .PAGE_WIDTH, 1);
1576 2508 2
1577 2509 2 ! re-init
1578 2510 2 STRING_DESC[SIZE] = K_MAX_BUFFER_SIZE; ! reset buffer size
1579 2511 2 GET_JOB_DESCRIPTION (.SCB,
1580 2512 2 STRING_DESC[0], ! SCB addr.
1581 2513 2 ! Use present tense
1582 2514 2 STRING_DESC[SIZE]); ! Buffer descriptor
1583 2515 2 ! Returned length
1584 2516 2 : Job description - create a sentence describing the current job.
1585 2517 2 ! re-init
1586 2518 2 STRING_DESC[SIZE] = K_MAX_BUFFER_SIZE; ! reset buffer size
1587 2519 2 GET_JOB_DESCRIPTION (.SCB,
1588 2520 2 STRING_DESC[0], ! SCB addr.
1589 2521 2 ! Use present tense
1590 2522 2 STRING_DESC[SIZE]); ! Buffer descriptor
1591 2523 2 ! Returned length
1592 2524 2 RET_LEN[0] = RETURN_FRAME_LENGTH
1593 2525 2 (.SCB,
1594 2526 2 STRING_DESC[0], ! string ref.
1595 2527 2 PAGE_REF[0,0,.PAGE_WIDTH], ! ref to frame
1596 2528 2 .PAGE_WIDTH, ! cols to fill
1597 2529 2 .BOTTOM_OFFSET - .TOP_OFFSET); ! rows to fill
1598 2530 2
1599 2531 2 IF .RET_LEN[0] GTR 0
1600 2532 2 THEN
1601 2533 2 BOTTOM_OFFSET = .BOTTOM_OFFSET - (.RET_LEN[0] + 1);
1602 2534 2 ! offset before inserting
1603 2535 2 : insert the string delimited
1604 2536 2 INSERT_FRAME (.SCB,
1605 2537 2 STRING_DESC[0], ! string ref.
1606 2538 2 PAGE_REF[0,.BOTTOM_OFFSET,.PAGE_WIDTH], ! ref to frame
1607 2539 2 .PAGE_WIDTH, ! cols to fill
1608 2540 2 .RET_LEN[0]); ! rows to fill
1609 2541 2
1610 2542 2
1611 2543 2
1612 2544 2 ! User note
```

1613 2 !
1614 2 ! re-init
1615 2 STRING_DESC[SIZE] = K_MAX_BUFFER_SIZE; ! reset buffer size
1616 2
1617 2 ! Get the user note
1618 2 GET_USER_NOTE
1619 2 (.SCB,
1620 2 STRING_DESC[0],
1621 2 STRING_DESC[SIZE]); ! SCB addr.
1622 2 ! Buffer descriptor
1623 2 ! Returned length
1624 2
1625 2 RET_LEN[0] = RETURN_FRAME_LENGTH
1626 2 (.SCB,
1627 2 STRING_DESC[0],
1628 2 PAGE_REF[0,0..PAGE_WIDTH], ! string ref.
1629 2 .PAGE_WIDTH, ! ref to frame
1630 2 .BOTTOM_OFFSET - .TOP_OFFSET); ! cols to fill
1631 2 ! rows to fill
1632 2
1633 2 ! insert the string delimited
1634 2 INSERT_FRAME (.SCB,
1635 2 STRING_DESC[0],
1636 2 PAGE_REF[0..TOP_OFFSET..PAGE_WIDTH], ! string ref.
1637 2 .PAGE_WIDTH, ! ref to frame
1638 2 .RET_LEN[0]; ! cols to fill
1639 2 ! rows to fill
1640 2
1641 2 IF .RET_LEN[0] GTR 0
1642 2 THEN TOP_OFFSET = .TOP_OFFSET + .RET_LEN[0] + 1
1643 2 ELSE TOP_OFFSET = .TOP_OFFSET + 1; ! adjust & allow for spacing
1644 2
1645 2 ! User Name
1646 2
1647 2 ! re-init
1648 2 STRING_DESC[SIZE] = K_MAX_BUFFER_SIZE; ! reset buffer size
1649 2
1650 2 RET_LEN[0] = INSERT_NAME_BANNER (
1651 2 (.SCB,
1652 2 SCB_SIZE(USER NAME), ! user name descriptor
1653 2 PAGE_REF[0..TOP_OFFSET..PAGE_WIDTH],
1654 2 .PAGE_WIDTH, ! ref to frame
1655 2 .BOTTOM_OFFSET-.TOP_OFFSET, ! max width Bann
1656 2
1657 2 14); ! space left
1658 2 ! max hght Bann string desired
1659 2
1660 2 TOP_OFFSET = .TOP_OFFSET + .RET_LEN[0] + 2;! adjust for banner & spacing
1661 2
1662 2 ! Job Name
1663 2
1664 2 ! re-init
1665 2 STRING_DESC[SIZE] = K_MAX_BUFFER_SIZE; ! reset buffer size
1666 2
1667 2 GET_JOB_NAME
1668 2 (.SCB,
1669 2 STRING_DESC[0],
2601 2 STRING_DESC[SIZE]); ! SCB addr.
2600 2 ! Buffer descriptor
2601 2 ! Returned length

```

1670 2602 2
1671 2603 2 RET_LEN[0] = INSERT_NAME_BANNER (
1672 2604 2     .SCB,
1673 2605 2     STRING_DESC[SIZE], ! job name desc
1674 2606 2     PAGE_REF[0..TOP_OFFSET..PAGE_WIDTH], ! ref to frame
1675 2607 2     .PAGE_WIDTH, ! max width Bann
1676 2608 2     .BOTTOM_OFFSET-.TOP_OFFSET, ! space left
1677 2609 2     ?); ! max hght Bann str desired
1678 2610 2
1679 2611 2
1680 2612 2
1681 2613 2
1682 2614 2 IF .RET_LEN[0] GTR 0
1683 2615 2 THEN TOP_OFFSET = .TOP_OFFSET + .RET_LEN[0] + 2; ! adjust & allow for spacing
1684 2616 2
1685 2617 2 | Get and insert the filename banner
1686 2618 2
1687 2619 2
1688 2620 2
1689 2621 2 | re-init
1690 2622 2 STRING_DESC[SIZE] = K_MAX_BUFFER_SIZE; ! reset buffer size
1691 2623 2 IF (.BOTTOM_OFFSET - 9) GTR .TOP_OFFSET ! test for enough room
1692 2624 2 THEN
1693 2625 2 BEGIN
1694 2626 2     BOTTOM_OFFSET = .BOTTOM_OFFSET - 9; ! offset before inserting
1695 2627 2
1696 2628 2     INSERT_JOBNUMBER_BANNER
1697 2629 2         (.SCB,
1698 2630 2         STRING_DESC[0], ! Buffer desc.
1699 2631 2         PAGE_REF[0..BOTTOM_OFFSET..PAGE_WIDTH], ! ref to frame
1700 2632 2         .PAGE_WIDTH, ! max width Bann
1701 2633 2         ?); ! rows to fill
1702 2634 2
1703 2635 2
1704 2636 1 END;

```

00FC 00000 FILL_JOB_FLAG:

								.WORD	Save R2, R3, R4, R5, R6, R7	2454
								MOVAB	-516(SP), SP	2473
								MOVZWL	#512 STRING DESC	2474
								MOVAB	BUFFER, STRING_DESC+4	2476
								CLRL	TOP_OFFSET	2477
								SUBL3	#2, PAGE_LENGTH, BOTTOM_OFFSET	2483
								PUSHL	#3	
								MOVL	PAGE_WIDTH, R7	
								PUSHL	R7	
								MOVL	PAGE_REF, R4	
								MULL3	R7, TOP_OFFSET, R0	
								PUSHAB	(R0)[R4]	
								MOVL	SCB, R6	2482
								MOVZBL	678(R6), -(SP)	2483
								PUSHL	R6	
								CALLS	#5, FILL_FRAME	

SEPARATE
V04-001

Print Symbiont -- separation routines
FILL_JOB_FLAG - Insert Information in

E 1
B 16-Sep-1984 02:23:03
14-Sep-1984 22:32:26

VAX-11 Bliss-32 V4.0-742
[PRTSMB.SRC]SEPARATE.B32:2

Page 52
(16)

SEP
V04

50			03	DD	0003B	PUSHL	#3
	53		57	DD	0003D	PUSHL	R7
		6044	57	C5	0003F	MULL3	R7 BOTTOM_OFFSET, R0
	7E	02A6	C6	9A	00043	PUSHAB	(R0)[R4]
0000V	CF		56	DD	0004B	MOVZBL	678(R6), -(SP)
	6E	0200	05	FB	0004D	PUSHL	R6
		8F	3C	00052	CALLS	#5 FILL FRAME	
			5E	DD	00057	MOVZWL	#512, STRING_DESC
		04	AE	9F	00059	PUSHL	SP
0000V	CF		56	DD	0005C	PUSHAB	STRING_DESC
			03	FB	0005E	PUSHL	R6
			01	DD	00063	CALLS	#3, GET_SYSTEM_ANNOUNCEMENT
	50		57	DD	00065	PUSHL	#1
	50	01	A2	9E	00067	PUSHL	R7
	50		57	C4	0006B	MOVAB	1(R2) R0
		6044	9F	0006E	MULL2	R7 R0	
		OC	AE	9F	00071	PUSHAB	(R0)[R4]
0000V	CF		56	DD	00074	PUSHAB	STRING_DESC
	6E	0200	05	FB	00076	PUSHL	R6
		8F	3C	0007B	CALLS	#5 CENTER FRAME	
			5E	DD	00080	MOVZWL	#512, STRING_DESC
		04	AE	9F	00082	PUSHL	SP
0000V	CF		56	DD	00085	PUSHAB	STRING_DESC
			03	FB	00087	PUSHL	R6
			01	DD	0008C	CALLS	#3, GET_DIGITAL_LOGO
	50		57	DD	0008E	PUSHL	#1
	50	01	A3	9E	00090	PUSHL	R7
	50		57	C4	00094	MOVAB	1(R3) R0
		6044	9F	00097	MULL2	R7 R0	
		OC	AE	9F	0009A	PUSHAB	(R0)[R4]
0000V	CF		56	DD	0009D	PUSHAB	STRING_DESC
	52	0200	05	FB	0009F	PUSHL	R6
	6E		04	CO	000A4	CALLS	#5, CENTER FRAME
		8F	3C	000A7	ADDL2	#4, TOP_OFFSET	
			5E	DD	000AC	MOVZWL	#512, STRING_DESC
		04	AE	9F	000AE	PUSHL	SP
			01	DD	000B1	PUSHAB	STRING_DESC
			56	DD	000B3	PUSHL	#1
7E	0000V	CF	04	FB	000B5	CALLS	R6
	53		52	C3	000BA	SUBL3	#4, GET_JOB_DESCRIPTION
		0090	8F	BB	000BE	PUSHR	TOP_OFFSET, BOTTOM_OFFSET, -(SP)
		OC	AE	9F	000C2	PUSHAB	#^MZR4,R7>
0000V	CF		56	DD	000C5	PUSHL	STRING_DESC
	55	0200	05	FB	000C7	CALLS	R6
			50	DO	000CC	MOVL	#5, RETURN_FRAME_LENGTH
			08	15	000CF	BLEQ	R0, RET_LEN
50	53		55	C5	000D1	SUBL3	1\$
	53	FF	A0	9E	000D5	MOVAB	RET_LEN, BOTTOM_OFFSET, R0
			55	DD	000D9	PUSHL	-1(R0), BOTTOM_OFFSET
			57	DD	000DB	PUSHL	RET_LEN
50	53		57	C5	000D0	MULL3	R7 BOTTOM_OFFSET, R0
		6044	9F	000E1	PUSHAB	(R0)[R4]	
		OC	AE	9F	000E4	PUSHAB	STRING_DESC
0000V	CF		56	DD	000E7	PUSHL	R6
	6E	0200	05	FB	000E9	CALLS	#5, INSERT_FRAME
		8F	3C	000EE	MOVZWL	#512, STRING_DESC	

SEPARATE
V04-001

Print_Symbiont -- separation routines
FILL_JOB_FLAG - Insert Information in

F 1
16-Sep-1984 02:23:03
14-Sep-1984 22:32:26

VAX-11 Bliss-32 V4.0-742
[PRTSMB.SRC]SEPARATE.B32;2

Page 53
(16)

SEP
V04

			SE	DD	000F3	PUSHL	SP
			AE	9F	000F5	PUSHAB	STRING_DESC
			56	DD	000F8	PUSHL	R6
			03	FB	000FA	CALLS	#3. GET_USER_NOTE
			52	C3	000FF	SUBL3	TOP_OFFSET, BOTTOM_OFFSET, -(SP)
			8F	BB	00103	PUSHR	#^MZR4,R7>
			AE	9F	00107	PUSHAB	STRING_DESC
			56	DD	0010A	PUSHL	R6
			05	FB	0010C	CALLS	#5. RETURN_FRAME_LENGTH
			50	DO	00111	MOVL	R0, RET_LEN
			55	DD	00114	PUSHL	RET_LEN
			57	DL	00116	PUSHL	R7
			57	CS	00118	MULL3	R7, TOP_OFFSET, R0
			6044	9F	0011C	PUSHAB	(R0)[R4]
			0C	AE	0011F	PUSHAB	STRING_DESC
			56	DD	00122	PUSHL	R6
			05	FB	00124	CALLS	#5. INSERT_FRAME
			55	D5	00129	TSTL	RET_LEN
			07	15	0012B	BLEQ	2\$
			52	A542	9E	MOVAB	1(RET_LEN)[TOP_OFFSET], TOP_OFFSET
			02	11	00132	BRB	3\$
			52	D6	00134	INCL	TOP_OFFSET
			6E	0200	8F	MOVZWL	#512, STRING_DESC
			0E	3C	00136	PUSHL	#14
			52	C3	0013B	SUBL3	TOP_OFFSET, BOTTOM_OFFSET, -(SP)
			57	DD	00141	PUSHL	R7
			57	C5	00143	MULL3	R7, TOP_OFFSET, R0
			6044	9F	00147	PUSHAB	(R0)[R4]
			016C	C6	0014A	PUSHAB	364(R6)
			56	DD	0014E	PUSHL	R6
			06	FB	00150	CALLS	#6. INSERT_NAME_BANNER
			55	50	00155	MOVL	R0, RET_LEN
			52	A542	9E	MOVAB	2(RET_LEN)[TOP_OFFSET], TOP_OFFSET
			6E	0200	8F	MOVZWL	#512, STRING_DESC
			02	3C	0015D	PUSHL	SP
			56	DD	00162	PUSHAB	STRING_DESC
			04	AE	00164	PUSHL	R6
			56	DD	00167	CALLS	#3. GET_JOB_NAME
			03	FB	00169	PUSHL	#7
			07	DD	0016E	SUBL3	TOP_OFFSET, BOTTOM_OFFSET, -(SP)
			52	C3	00170	PUSHL	R7
			57	DD	00174	MULL3	R7, TOP_OFFSET, R0
			57	C5	00176	PUSHAB	(R0)[R4]
			6044	9F	0017A	PUSHAB	STRING_DESC
			10	AE	0017D	PUSHL	R6
			56	DD	00180	CALLS	#6. INSERT_NAME_BANNER
			06	FB	00182	MOVL	R0, RET_LEN
			55	50	00187	BLEQ	4\$
			05	15	0018A	MOVAB	2(RET_LEN)[TOP_OFFSET], TOP_OFFSET
			52	A542	9E	MOVZWL	#512, STRING_DESC
			02	3C	00191	-9(R3), R0	-9(R3), R0
			6E	0200	8F	MOVAB	R0, TOP_OFFSET
			F7	A3	9E	CPL	5\$
			50	D1	0019A	BLEQ	#9. BOTTOM_OFFSET
			17	15	0019D	SUBL2	#7
			53	09	C2	PUSHL	R7
			07	DD	001A2	PUSHL	R7, R3
			57	DD	001A4	PUSHL	
			57	C4	001A6	MULL2	

SEPARATE
V04-001

Print Symbiont -- separation routines
FILL_JOB_FLAG - Insert Information into the JOB

G 1

16-Sep-1984 02:23:03
14-Sep-1984 22:32:26

VAX-11 Bliss-32 V4.0-742
[PRTSMB.SRC]SEPARATE.B32;2

Page 54
(16)

OC	6344	9F	001A9	PUSHAB	(R3)[R4]
	AE	9F	001AC	PUSHAB	STRING_DESC
	56	DD	001AF	PUSHL	R6
0000V	CF	05	FB 001B1	CALLS	#5, INSERT_JOBNUMBER_BANNER
		04	001B6 5\$:	RET	

: 2630
: 2631
: 2636

: Routine Size: 439 bytes. Routine Base: CODE + 098F

SEP
V04

```
1706      2637 1 %sbttl 'FILL_JOB_TRAILER - Insert Information into the JOB Page'
1707      2638 1 !++
1708      2639 1 : Functional Description:
1709      2640 1 : This procedure controls all inserts required for the JOB Page.
1710      2641 1 :
1711      2642 1 : Formal Parameters:
1712      2643 1 :   SCB           - Address of the SCB
1713      2644 1 :   PAGE_REF     - Pointer to the Page (first byte)
1714      2645 1 :   PAGE_LENGTH   - Length of Frame
1715      2646 1 :   PAGE_WIDTH    - Width of Frame
1716
1717      2648 1 : Implicit Inputs:
1718      2649 1 :   none
1719
1720      2650 1 : Implicit Outputs:
1721      2652 1 :   none
1722
1723      2653 1 : Returned Value:
1724      2655 1 :   none
1725
1726      2657 1 : Side Effects:
1727      2658 1 :   none
1728      2659 1 !--
1729      2660 1 ROUTINE FILL_JOB_TRAILER (
1730      2661 1 :   SCB           : REF SBBLOCK,
1731      2662 1 :   PAGE_REF     : REF PAGE_ARRAY,
1732      2663 1 :   PAGE_WIDTH,
1733      2664 1 :   PAGE_LENGTH
1734      2665 1 : ) : NOVALUE =
1735      2666 2 BEGIN
1736
1737      2668 2 LITERAL K_MAX_BUFFER_SIZE = 512;
1738
1739      2669 2 LOCAL
1740      2671 2 :   RET_LEN       : VECTOR[1],
1741      2672 2 :   RET_WIDE      : VECTOR[1],
1742      2673 2 :   RIGHT_OFFSET,
1743      2674 2 :   LEFT_OFFSET ,
1744      2675 2 :   TOP_OFFSET ,
1745      2676 2 :   BOTTOM_OFFSET,
1746      2677 2 :   BUFFER        : VECTOR [512,byte],          ! Assume max size 512 bytes
1747      2678 2 :   STRING_DESC : VECTOR [2];                      ! Descriptor to current string
1748
1749      2680 2 : Allocate the buffer for "GET_xxx" Routines
1750
1751      2682 2 STRING_DESC[SIZE] = XALLOCATION(BUFFER);          ! allocate for routines
1752      2683 2 STRING_DESC[ADDR] = BUFFER;                      ! init address
1753
1754      2685 2 : Top of page
1755
1756      2687 2 TOP_OFFSET = 0;
1757      2688 2 BOTTOM_OFFSET = .PAGE_LENGTH;
1758
1759      2690 2 : insert the burst characters
1760      2691 2 :   FILL_FRAME (.SCB
1761      2692 2 :     .SCB[PSMSB_JOB_BURST_CHAR],
1762      2693 2 :     PAGE_REF[0-.TOP_OFFSET+2,.PAGE_WIDTH], .PAGE_WIDTH, 3);
```

```
1763 2694 2 ! re-init
1764 2695 2 STRING_DESC[SIZE] = K_MAX_BUFFER_SIZE; ! reset buffer size
1765 2696 2
1766 2697 2
1767 2698 2 GET_EOJ
1768 2699 2 (.SCB,
1769 2700 2 STRING_DESC[0], ! Buffer descriptor
1770 2701 2 STRING_DESC[SIZE]); ! Returned length
1771 2702 2
1772 2703 2 RET_LEN[0] = INSERT_NAME_BANNER (
1773 2704 2 .SCB,
1774 2705 2 STRING_DESC[SIZE], ! eojo name desc
1775 2706 2 PAGE_REF[0..TOP_OFFSET..PAGE_WIDTH], ! ref to frame
1776 2707 2 .PAGE_WIDTH, ! max width Bann
1777 2708 2 .BOTTOM_OFFSET - .TOP_OFFSET, ! frame size
1778 2709 2
1779 2710 2 7); ! max hght Bann str
1780 2711 2
1781 2712 2
1782 2713 2 IF .RET_LEN[0] GTR 0
1783 2714 2 THEN
1784 2715 2 TOP_OFFSET = .TOP_OFFSET + .RET_LEN[0] + 2; ! adjust & allow for spacing
1785 2716 2 ! includes the burst also...
1786 2717 2 ! two spaces...
1787 2718 2
1788 2719 2 ! Bottom of page
1789 2720 2
1790 2721 2 BOTTOM_OFFSET = .PAGE_LENGTH - 5; ! offset includes burst offset
1791 2722 2
1792 2723 2 FILL_FRAME (.SCB,
1793 2724 2 .SCB[PSMSB_JOB_BURST_CHAR],
1794 2725 2 PAGE_REF[0..BOTTOM_OFFSET..PAGE_WIDTH], .PAGE_WIDTH,3);
1795 2726 2
1796 2727 2 ! re-init
1797 2728 2 STRING_DESC[SIZE] = K_MAX_BUFFER_SIZE; ! reset buffer size
1798 2729 2
1799 2730 2 ! Get the digital logo and output to page ... assume not greater than
1800 2731 2 ! amount allocated.... truncation otherwise occurs
1801 2732 2
1802 2733 2 GET_DIGITAL_LOGO
1803 2734 2 (.SCB,
1804 2735 2 STRING_DESC[0], ! SCB addr.
1805 2736 2 STRING_DESC[SIZE]); ! Buffer descriptor
1806 2737 2 ! Returned length
1807 2738 2
1808 2739 2 CENTER_FRAME (.SCB,
1809 2740 2 STRING_DESC[0],
1810 2741 2 PAGE_REF[0..BOTTOM_OFFSET+1..PAGE_WIDTH], .PAGE_WIDTH, 1);
1811 2742 2
1812 2743 2
1813 2744 2 ! re-init
1814 2745 2 STRING_DESC[SIZE] = K_MAX_BUFFER_SIZE; ! reset buffer size
1815 2746 2
1816 2747 2
1817 2748 2 GET_RULER_COARSE
1818 2749 2 (.SCB,
1819 2750 2 STRING_DESC[0], ! SCB addr.
          STRING_DESC[SIZE]); ! Buffer descriptor
          ! Returned length
```

```

1820 2751 2 SCROLL_FRAME (.SCB,
1821 2752 2 STRING_DESC[0]
1822 2753 2 PAGE_REF[0,.BOTTOM_OFFSET+4,.PAGE_WIDTH], .PAGE_WIDTH, 1);
1823 2754 2
1824 2755 2 SCROLL_FRAME (.SCB,
1825 2756 2 $DESCRIPTOR ('1234567890'),
1826 2757 2 PAGE_REF[0,.BOTTOM_OFFSET+5,.PAGE_WIDTH], .PAGE_WIDTH, 1);
1827 2758 2
1828 2759 2 ! Create a sentence describing the current job.
1829 2760 2
1830 2761 2 ! re-init
1831 2762 2 STRING_DESC[SIZE] = K_MAX_BUFFER_SIZE; ! reset buffer size
1832 2763 2
1833 2764 2 GET_JOB_DESCRIPTION
1834 2765 2 (.SCB, ! SCB addr.
1835 2766 2 0 ! Use past tense
1836 2767 2 STRING_DESC[0], ! Buffer descriptor
1837 2768 2 STRING_DESC[SIZE]); ! Returned length
1838 2769 2
1839 2770 2 RET_LEN[0] = RETURN_FRAME_LENGTH
1840 2771 2 (.SCB, ! string ref.
1841 2772 2 STRING_DESC[0], ! ref to frame
1842 2773 2 PAGE_REF[0,0,.PAGE_WIDTH], ! cols to fill
1843 2774 2 .PAGE_WIDTH, ! rows to fill
1844 2775 2 .BOTTOM_OFFSET - .TOP_OFFSET); ! rows to fill
1845 2776 2
1846 2777 2 IF .RET_LEN[0] GTR 0
1847 2778 2 THEN
1848 2779 2 BOTTOM_OFFSET = .BOTTOM_OFFSET - (.RET_LEN[0] + 1); ! adjust & allow for spacing
1849 2780 2 ! before inserting
1850 2781 2
1851 2782 2 ! insert the string delimited
1852 2783 2 INSERT_FRAME (.SCB,
1853 2784 2 STRING_DESC[0], ! string ref.
1854 2785 2 PAGE_REF[0,.BOTTOM_OFFSET,.PAGE_WIDTH], ! ref to frame
1855 2786 2 .PAGE_WIDTH, ! cols to fill
1856 2787 2 .RET_LEN[0]); ! rows to fill
1857 2788 2
1858 2789 2
1859 2790 2
1860 2791 2 ! User name
1861 2792 2
1862 2793 2 ! User name
1863 2794 2 RET_LEN[0] = INSERT_NAME_BANNER
1864 2795 2 (.SCB, ! user name desc
1865 2796 2 SCB_SIZE (USER NAME), ! PAGE_REF[0,.TOP_OFFSET,.PAGE_WIDTH],
1866 2797 2 PAGE_REF[0,.TOP_OFFSET,.PAGE_WIDTH], ! ref to frame
1867 2798 2 .PAGE_WIDTH, ! max width Bann
1868 2799 2 .BOTTOM_OFFSET - .TOP_OFFSET, ! frame size
1869 2800 2 ?); ! max hght Bann str desired
1870 2801 2
1871 2802 2
1872 2803 2 ! re-init
1873 2804 2 IF .RET_LEN[0] GTR 0
1874 2805 2 THEN
1875 2806 2 TOP_OFFSET = .TOP_OFFSET + .RET_LEN[0] + 2; ! adjust & allow for spacing
1876 2807 2

```

1877 2808 2 ! Job name
1878 2809 22 !
1879 2810 22 ! re-init
1880 2811 22 STRING_DESC[SIZE] = K_MAX_BUFFER_SIZE; ! reset buffer size
1881 2812 22
1882 2813 22 GET_JOB_NAME (.SCB,
1883 2814 22 STRING_DESC[0],
1884 2815 22 STRING_DESC[SIZE]); ! Buffer descriptor
1885 2816 22 ! Returned length
1886 2817 22
1887 2818 22 RET_LEN[0] = INSERT_NAME_BANNER (
1888 2819 22 .SCB,
1889 2820 22 STRING_DESC[SIZE], ! job name size
1890 2821 22 PAGE_REF[0,.TOP_OFFSET,.PAGE_WIDTH], ! ref to frame
1891 2822 22 .PAGE_WIDTH, ! max width Bann
1892 2823 22 .BOTTOM_OFFSET - .TOP_OFFSET, ! frame size
1893 2824 22 7); ! max hght Bann str
1894 2825 22
1895 2826 22
1896 2827 22
1897 2828 22 ! re-init
1898 2829 22 IF .RET_LEN[0] GTR 0
1899 2830 22 THEN
1900 2831 22 TOP_OFFSET = .TOP_OFFSET + .RET_LEN[0] + 2; ! adjust & allow for spacing
1901 2832 22
1902 2833 22 ! Receipt box
1903 2834 22
1904 2835 22 RIGHT_OFFSET = .PAGE_WIDTH; ! right_offset and left_offset
1905 2836 22 LEFT_OFFSET = 0; ! are positional offsets for
1906 2837 22 a specific frame_length and
1907 2838 22 range.
1908 2839 22
1909 2840 22 ! re-init
1910 2841 22 STRING_DESC[SIZE] = K_MAX_BUFFER_SIZE; ! reset buffer size
1911 2842 22
1912 2843 22 GET_RECEIPT_BOX (.SCB,
1913 2844 22 STRING_DESC[0],
1914 2845 22 STRING_DESC[SIZE]); ! Buffer descriptor
1915 2846 22 ! Returned length
1916 2847 22 ! Get the width needed for insert (assume length of seven)
1917 2848 22 RET_WIDE[0] = RETURN_FRAME_WIDTH
1918 2849 22 (.SCB,
1919 2850 22 STRING_DESC[0], ! string ref.
1920 2851 22 PAGE_REF[0,.TOP_OFFSET,.PAGE_WIDTH], ! ref to frame
1921 2852 22 .RIGHT_OFFSET -.LEFT_OFFSET, ! special width
1922 2853 22 8); ! rows to fill
1923 2854 22
1924 2855 22
1925 2856 22 RET_LEN[0] = RETURN_FRAME_LENGTH
1926 2857 22 (.SCB,
1927 2858 22 STRING_DESC[0], ! string ref.
1928 2859 22 PAGE_REF[0,0,.PAGE_WIDTH], ! ref to frame
1929 2860 22 .RET_WIDE[0], ! cols to fill
1930 2861 22 .BOTTOM_OFFSET - .TOP_OFFSET); ! rows to fill
1931 2862 22
1932 2863 22 IF .RET_LEN[0] GTR 0
1933 2864 22 THEN

```

1934 2865 2 BOTTOM_OFFSET = .BOTTOM_OFFSET - (.RET_LEN[0] + 1);
1935 2866 2 ; adjust & allow for spacing
1936 2867 2 before inserting
1937 2868 2 RIGHT_OFFSET = .RIGHT_OFFSET - .RET_WIDE[0];: offset before inserting
1938 2869 2
1939 2870 2 MOVE_FRAME (.SCB,
1940 2871 2     STRING_DESC[0], ! string frame reference
1941 2872 2     PAGE_REF[.RIGHT_OFFSET,.BOTTOM_OFFSET,.PAGE_WIDTH],
1942 2873 2 ; ref to frame
1943 2874 2     .RET_WIDE[0], ! width
1944 2875 2     .RET_LEN[0]); ! rows to fill
1945 2876 2
1946 2877 2 ; Get and insert the filename banner
1947 2878 2
1948 2879 2
1949 2880 2 ; re-init
1950 2881 2 STRING_DESC[SIZE] = K_MAX_BUFFER_SIZE; ! reset buffer size
1951 2882 2
1952 2883 2 INSERT_JOBNUMBER_BANNER
1953 2884 2     (.SCB,
1954 2885 2     STRING_DESC[0], ! Buffer desc.
1955 2886 2     PAGE_REF[.LEFT_OFFSET,.BOTTOM_OFFSET,.PAGE_WIDTH],
1956 2887 2 ; ref to frame
1957 2888 2     .RIGHT_OFFSET-.LEFT_OFFSET, ! max width Bann
1958 2889 2     .BOTTOM_OFFSET - .TOP_OFFSET); ! rows to fill
1959 2890 2
1960 2891 2 ; re-init
1961 2892 2 STRING_DESC[SIZE] = K_MAX_BUFFER_SIZE; ! reset buffer size
1962 2893 2 RIGHT_OFFSET = .PAGE_WIDTH; ! right_offset and left_offset
1963 2894 2 LEFT_OFFSET = 0; ! are positional offsets for
1964 2895 2 ; a specific frame_length and
1965 2896 2 ; range.
1966 2897 2
1967 2898 2 ; GET_ACCOUNTING_INFO
1968 2899 2     (.SCB,
1969 2900 2     STRING_DESC[0], ! Buffer descriptor
1970 2901 2     STRING_DESC[SIZE]); ! Returned length
1971 2902 2
1972 2903 2 ; insert the string delimited
1973 2904 2     INSERT_FRAME (.SCB,
1974 2905 2     STRING_DESC[0], ! string ref.
1975 2906 2     PAGE_REF[0, 45, .PAGE_WIDTH], ! ref to frame
1976 2907 2     .PAGE_WIDTH, ! cols to fill
1977 2908 2     2); ! rows to fill
1978 2909 1 END;

```

30 39 38 37 36 35 34 33 32 31 00B46 P.AAF: .ASCII \1234567890\
 0000000A 00B50 P.AAE: .LONG 10
 00000000 00B54 .ADDRESS P.AAF

OFFC 00000 FILL_JOB_TRAILER:
 5B 0000V CF 9E 00002 :WORD Save R2,R3,R4,R5,R6,R7,R8,R9,R10,R11
 MOVAB INSERT_NAME_BANNER, R11 : 2660

SEPARATE
V04-001Print Symbiont -- separation routines
FILL_JOB_TRAILER - Insert Information into theM 1
16-Sep-1984 02:23:03
14-Sep-1984 22:32:26VAX-11 Bliss-32 V4.0-742
[PRTSMB.SRC]SEPARATE.B32;2Page 60
(17)SEP
V04

		SE	FDFC	CE	9E 00007	MOVAB	-516(SP), SP	
	04	7E	0200	8F	3C 0000C	MOVZWL	#512, STRING_DESC	2682
			08	AE	9E 00011	MOVAB	BUFFER, STRING_DESC+4	2683
				52	D4 00016	CLRL	TOP_OFFSET	2687
		53	10	AC	DD 00018	MOVL	PAGE_LENGTH, BOTTOM_OFFSET	2688
				03	DD 0001C	PUSHL	#3	2693
		57	0C	AC	DD 0001E	MOVL	PAGE_WIDTH, R7	
				57	DD 00022	PUSHL	R7	
		54	08	AC	DD 00024	MOVL	PAGE_REF, R4	
		50	02	A2	9E 00028	MOVAB	2(R2), R6	
		50		57	C4 0002C	MULL2	R7, R6	
				6044	9F 0002F	PUSHAB	(R6)[R4]	
		56	04	AC	DD 00032	MOVL	SCB, R6	2692
		7E	02A6	C6	9A 00036	MOVZBL	678(R6), -(SP)	2693
	0000V	CF		56	DD 0003B	PUSHL	R6	
		6E	0200	05	FB 0003D	CALLS	#5, FILL_FRAME	
				8F	3C 00042	MOVZWL	#512, STRING_DESC	2696
				5E	DD 00047	PUSHL	SP	2701
				04	AE 9F 00049	PUSHAB	STRING_DESC	2700
					56 DD 0004C	PUSHL	R6	2699
	0000V	CF		03	FB 0004E	CALLS	#3, GET_EOJ	
				07	DD 00053	PUSHL	#7	
	7E	53		52	C3 00055	SUBL3	TOP_OFFSET, BOTTOM_OFFSET, -(SP)	2706
				57	DD 00059	PUSHL	R7	2709
	50	52		57	C5 0005B	MULL3	R7, TOP_OFFSET, R0	2708
				6044	9F 0005F	PUSHAB	(R6)[R4]	2706
			10	AE	9F 00062	PUSHAB	STRING_DESC	2705
				56	DD 00065	PUSHL	R6	2706
		68		06	FB 00067	CALLS	#6, INSERT_NAME_BANNER	
		55		50	DD 0006A	MOVL	R0, RET_LEN	
				05	15 0006D	BLEQ	1\$	
	53	10	S2	02 A542	9E 0006F	MOVAB	2(RET_LEN)[TOP_OFFSET], TOP_OFFSET	2713
			AC	05	C3 00074	SUBL3	#5, PAGE_LENGTH, BOTTOM_OFFSET	2715
				03	DD 00079	PUSHL	#3	2721
				57	DD 0007B	PUSHL	R7	2725
	50	53		57	C5 0007D	MULL3	R7, BOTTOM_OFFSET, R0	
				6044	9F 00081	PUSHAB	(R6)[R4]	
		7E	02A6	C6	9A 00084	MOVZBL	678(R6), -(SP)	
	0000V	CF		56	DD 00089	PUSHL	R6	
		6E	0200	05	FB 0008B	CALLS	#5, FILL_FRAME	2728
				8F	3C 00090	MOVZWL	#512, STRING_DESC	2736
				5E	DD 00095	PUSHL	SP	2735
				04	AE 9F 00097	PUSHAB	STRING_DESC	2734
	0000V	CF		56	DD 0009A	PUSHL	R6	
				03	FB 0009C	CALLS	#3, GET_DIGITAL_LOGO	
				01	DD 000A1	PUSHL	#1	2740
				57	DD 000A3	PUSHL	R7	
	50	50	01	A3	9E 000A5	MOVAB	1(R3), R0	
				57	C4 000A9	MULL2	R7, R6	
				6044	9F 000AC	PUSHAB	(R6)[R4]	
			OC	AE	9F 000AF	PUSHAB	STRING_DESC	2739
	0000V	CF		56	DD 000B2	PUSHL	R6	2740
		6E	0200	05	FB 000B4	CALLS	#5, CENTER_FRAME	2744
				8F	3C 000B9	MOVZWL	#512, STRING_DESC	2749
				5E	DD 000BE	PUSHL	SP	2748
				04	AE 9F 000C0	PUSHAB	STRING_DESC	2747
				56	DD 000C3	PUSHL	R6	

SEPARATE
V04-001

Print Symbiont -- separation routines
FILL_JOB_TRAILER - Insert Information

N 1
16-Sep-1984 02:23:03
14-Sep-1984 22:32:26

VAX-11 Bliss-32 V4.0-742
[PRTSMB.SRC]SEPARATE.B32:2

Page 61
(17)

SEP
V04

0000V	CF	03	FB	000C5	CALLS	#3. GET_RULER_COARSE	2753
		01	DD	000CA	PUSHL	#1	
50		57	DD	000CC	PUSHL	R7	
50	04	A3	9E	000CE	MOVAB	4(R3), R0	
		57	C4	000D2	MULL2	R7 R6	
		6044	9F	000D3	PUSHAB	(R6)[R4]	
	0C	AE	9F	000D8	PUSHAB	STRING_DESC	
0000V	CF	56	DD	000DB	PUSHL	R6	
		05	FB	000DD	CALLS	#5. SCROLL_FRAME	
		01	DD	000E2	PUSHL	#1	
50		57	DD	000E4	PUSHL	R7	
50	05	A3	9E	000E6	MOVAB	5(R3), R0	
		57	C4	000EA	MULL2	R7 R6	
		6044	9F	000ED	PUSHAB	(R6)[R4]	
	FF04	CF	9F	000FO	PUSHAB	P.AAE	
0000V	CF	56	DD	000F4	PUSHL	R6	
6E	0200	05	FB	000F6	CALLS	#5. SCROLL_FRAME	
		8F	3C	000FB	MOVZWL	#512. STRING_DESC	
		5E	DD	00100	PUSHL	SP	
	04	AE	9F	00102	PUSHAB	STRING_DESC	
		7E	D4	00105	CLRL	-(SP)	
		56	DD	00107	PUSHL	R6	
7E	CF	04	FB	00109	CALLS	#4. GET_JOB_DESCRIPTION	
53		52	C3	0010E	SUBL3	TOP_OFFSET, BOTTOM_OFFSET, -(SP)	
	0090	8F	BB	00112	PUSHR	#^MZR4,R7>	
	OC	AE	9F	00116	PUSHAB	STRING_DESC	
0000V	CF	56	DD	00119	PUSHL	R6	
55		05	FB	0011B	CALLS	#5. RETURN_FRAME_LENGTH	
		50	DD	00120	MOVL	R0, RET_LEN	
50	53	08	15	00123	BLEQ	2\$	
53	53	FF	55	C3 00125	SUBL3	RET_LEN, BOTTOM_OFFSET, R0	
		A0	9E	00129	MOVAB	-1(R0), BOTTOM_OFFSET	
		55	DD	0012D	PUSHL	RET_LEN	
50	53	57	DD	0012F	PUSHL	R7	
		57	C5	00131	MULL3	R7, BOTTOM_OFFSET, R0	
		6044	9F	00135	PUSHAB	(R6)[R4]	
	0C	AE	9F	00138	PUSHAB	STRING_DESC	
0000V	CF	56	DD	0013B	PUSHL	R6	
		05	FB	0013D	CALLS	#5. INSERT_FRAME	
7E	53	07	DD	00142	PUSHL	#7	
		52	C3	00144	SUBL3	TOP_OFFSET, BOTTOM_OFFSET, -(SP)	
50	52	57	DD	00148	PUSHL	R7	
		57	C5	0014A	MULL3	R7, TOP_OFFSET, R0	
		6044	9F	0014E	PUSHAB	(R6)[R4]	
	016C	C6	9F	00151	PUSHAB	364(R6)	
6B	55	56	DD	00155	PUSHL	R6	
		06	FB	00157	CALLS	#6. INSERT_NAME_BANNER	
55		50	DD	0015A	MOVL	R0, RET_LEN	
		05	15	0015D	BLEQ	3\$	
52	02	A542	9E	0015F	MOVAB	2(RET_LEN)[TOP_OFFSET], TOP_OFFSET	
6E	0200	8F	3C	00164	MOVZWL	#512. STRING_DESC	
	04	5E	DD	00169	PUSHL	SP	
		AE	9F	0016B	PUSHAB	STRING_DESC	
0000V	CF	56	DD	0016E	PUSHL	R6	
		03	FB	00170	CALLS	#3. GET_JOB_NAME	
		07	DD	00175	PUSHL	#7	
7E	53	52	C3	00177	SUBL3	TOP_OFFSET, BOTTOM_OFFSET, -(SP)	

SEPARATE
V04-001

Print Symbiont -- separation routines
FILL_JOB_TRAILER - Insert Information

B 2
16-Sep-1984 02:23:03
14-Sep-1984 22:32:26

VAX-11 Bliss-32 V4.0-742
[PRTSMB.SRC]SEPARATE.B32:2

Page 62
(17)

SEP.
V04

50		52	57	DD	0017B	PUSHL	R7		
			57	C5	0017D	MULL3	R7, TOP_OFFSET, R0		
		10	6044	9F	00181	PUSHAB	(R0)[R4]		
			AE	9F	00184	PUSHAB	STRING_DESC		
			56	DD	00187	PUSHL	R6		
			06	FB	00189	CALLS	#6. INSERT NAME_BANNER		
			50	DO	0018C	MOVL	R0, RET_LEN		
			05	15	0018F	BLEQ	48		
		52	02	A542	9E	00191	MOVAB	2(RET_LEN)[TOP_OFFSET], TOP_OFFSET	
		58			57	DO	00196	MOVL	R7, RIGHT_OFFSET
		6E	0200	8F	3C	0019B	CLRL	LEFT_OFFSET	
				59	D4	00199	MOVZWL	#512, STRING_DESC	
			04	5E	DD	001A0	PUSHL	SP	
				AE	9F	001A2	PUSHAB	STRING_DESC	
	0000V	CF		56	DD	001A5	PUSHL	R6	
				03	FB	001A7	CALLS	#3. GET_RECEIPT_BOX	
				08	DD	001AC	PUSHL	#8	
7E		58		59	C3	001AE	SUBL3	LEFT_OFFSET, RIGHT_OFFSET, -(SP)	
50		52		57	C5	001B2	MULL3	R7, TOP_OFFSET, R0	
			OC	6044	9F	001B6	PUSHAB	(R0)[R4]	
				AE	9F	001B9	PUSHAB	STRING_DESC	
	0000V	CF		56	DD	001BC	PUSHL	R6	
		5A		05	FB	001BE	CALLS	#5, RETURN_FRAME_WIDTH	
		53		50	DO	001C3	MOVL	RO, RET_WIDE	
7E			0410	52	C3	001C6	SUBL3	TOP_OFFSET, BOTTOM_OFFSET, -(SP)	
			OC	8F	BB	001CA	PUSHR	#^MZR4,R10>	
				AE	9F	001CE	PUSHAB	STRING_DESC	
	0000V	CF		56	DD	001D1	PUSHL	R6	
		55		05	FB	001D3	CALLS	#5, RETURN_FRAME_LENGTH	
				50	DO	001D8	MOVL	RO, RET_LEN	
50		53		08	15	001DB	BLEQ	58	
		53		55	C3	001DD	SUBL3	RET_LEN, BOTTOM_OFFSET, R0	
		58	FF	A0	9E	001E1	MOVAB	-1(R0), BOTTOM_OFFSET	
				5A	C2	001E5	SUBL2	RET_WIDE, RIGHT_OFFSET	
				55	DD	001E8	PUSHL	RET_LEN	
55		53		5A	DD	001EA	PUSHL	RET_WIDE	
50		55		57	C5	001EC	MULL3	R7, BOTTOM_OFFSET, R5	
				58	C1	001FO	ADDL3	RIGHT_OFFSET, R5, RO	
			OC	6044	9F	001F4	PUSHAB	(R0)[R4]	
				AE	9F	001F7	PUSHAB	STRING_DESC	
	0000V	CF		56	DD	001FA	PUSHL	R6	
		6E	0200	05	FB	001FC	CALLS	#5, MOVE FRAME	
7E		53		8F	3C	00201	MOVZWL	#512, STRING DESC	
7E		58		52	C3	00206	SUBL3	TOP_OFFSET, BOTTOM_OFFSET, -(SP)	
50		55		59	C3	0020A	SUBL3	LEFT_OFFSET, RIGHT_OFFSET, -(SP)	
				59	C1	0020E	ADDL3	LEFT_OFFSET, R5, RO	
			OC	6044	9F	00212	PUSHAB	(R0)[R4]	
				AE	9F	00215	PUSHAB	STRING_DESC	
	0000V	CF		56	DD	00218	PUSHL	R6	
		6E	0200	05	FB	0021A	CALLS	#5, INSERT_JOBNUMBER_BANNER	
		58		8F	3C	0021F	MOVZWL	#512, STRING DESC	
				57	DO	00224	MOVL	R7, RIGHT_OFFSET	
				59	D4	00227	CLRL	LEFT_OFFSET	
				04	00229	RET			

; Routine Size: 554 bytes, Routine Base: CODE + 0B58

SEPARATE
V04-001

Print Symbiont -- separation routines
FILL_JOB_TRAILER - Insert Information into the

C 2
16-Sep-1984 02:23:03
14-Sep-1984 22:32:26

VAX-11 Bliss-32 v4.0-742
[PRTSMB.SRC]SEPARATE.B32;2

Page 63
(17)

SEP
V04

18

1980 2910 1 %sbttl 'FILL_FILE_TRAILER - Insert Information into the FILE Page'
1981 2911 1 ++
1982 2912 1 Functional Description:
1983 2913 1 This procedure controls all inserts required for the FILE Page.
1984 2914 1
1985 2915 1 Formal Parameters:
1986 2916 1 SCB = Address of the SCB
1987 2917 1 PAGE_REF = Pointer to the Page (first byte)
1988 2918 1 PAGE_LENGTH = Length of Frame
1989 2919 1 PAGE_WIDTH = Width of Frame
1990 2920 1
1991 2921 1 Implicit Inputs:
1992 2922 1 none
1993 2923 1
1994 2924 1 Implicit Outputs:
1995 2925 1 none
1996 2926 1
1997 2927 1 Returned Value:
1998 2928 1 none
1999 2929 1
2000 2930 1 Side Effects:
2001 2931 1 none
2002 2932 1 --
2003 2933 1 ROUTINE FILL_FILE_TRAILER {
2004 2934 1 SCB : REF \$BLOCK,
2005 2935 1 PAGE_REF : REF PAGE_ARRAY,
2006 2936 1 PAGE_WIDTH,
2007 2937 1 PAGE_LENGTH
2008 2938 1): NOVALUE =
2009 2939 2 BEGIN
2010 2940 2
2011 2941 2 LITERAL K_MAX_BUFFER_SIZE = 512;
2012 2942 2
2013 2943 2 LOCAL
2014 2944 2 FORCE_LEN
2015 2945 2 RET_LEN : VECTOR[1],
2016 2946 2 TOP_OFFSET
2017 2947 2 BOTTOM_OFFSET,
2018 2948 2 BUFFER : VECTOR [512,byte]. ! Assume max size 512 bytes
2019 2949 2 STRING_DESC : VECTOR [2]; ! Descriptor to current string
2020 2950 2
2021 2951 2 ! Allocate the buffer for "GET_xxx" Routines
2022 2952 2
2023 2953 2 STRING_DESC[SIZE] = %ALLOCATION(BUFFER); ! allocate for routines
2024 2954 2 STRING_DESC[ADDR] = BUFFER; ! init address
2025 2955 2
2026 2956 2
2027 2957 2 ! Top of page
2028 2958 2
2029 2959 2 TOP_OFFSET = 0;
2030 2960 2 BOTTOM_OFFSET = .PAGE_LENGTH;
2031 2961 2
2032 2962 2 FILL_FRAME (.SCB,
2033 2963 2 .SCB[PSMSBFILE_BURST CHAR],
2034 2964 2 PAGE_REF[0..TOP_OFFSET+2..PAGE_WIDTH], .PAGE_WIDTH, 3);
2035 2965 2
2036 2966 2 FILL_FRAME (.SCB,

```

2037 2967 2      XC' '
2038 2968 2      PAGE_REF[10,.TOP_OFFSET+2,.PAGE_WIDTH], .PAGE_WIDTH-20, 3);
2039 2969 2
2040 2970 2      FILL_FRAME (.SCB,
2041 2971 2          .SCB[PSM$B_JOB_BURST_CHAR],
2042 2972 2          PAGE_REF[13, .TOP_OFFSET+2,.PAGE_WIDTH], .PAGE_WIDTH-26, 3);
2043 2973 2
2044 2974 2      ! re-init
2045 2975 2      STRING_DESC[SIZE] = K_MAX_BUFFER_SIZE;      ! reset buffer size
2046 2976 2
2047 2977 2      GET_EOF      (.SCB,
2048 2978 2          STRING_DESC[0],           ! Buffer descriptor
2049 2979 2          STRING_DESC[SIZE]);        ! Returned length
2050 2980 2
2051 2981 2      RET_LEN[0] = INSERT_NAME_BANNER (
2052 2982 2          .SCB,
2053 2983 2          STRING_DESC[SIZE],           ! file name desc
2054 2984 2          PAGE_REF[0,.TOP_OFFSET,.PAGE_WIDTH],       ! ref to frame
2055 2985 2          .PAGE_WIDTH,                 ! max width Bann ,
2056 2986 2          .BOTTOM_OFFSET - .TOP_OFFSET,           ! frame size
2057 2987 2          7);                      ! max hght Bann str
2058 2988 2
2059 2989 2
2060 2990 2
2061 2991 2      ! Adjust for the burst characters too
2062 2992 2      IF .RET_LEN[0] GTR 0
2063 2993 2      THEN
2064 2994 2          TOP_OFFSET      = .TOP_OFFSET + .RET_LEN[0] + 2;
2065 2995 2
2066 2996 2          ! adjust & allow for spacing
2067 2997 2          ! allow for two spaces...
2068 2998 2
2069 2999 2
2070 3000 2      BOTTOM_OFFSET = .PAGE_LENGTH - 5;           ! offset includes burst offset
2071 3001 2
2072 3002 2
2073 3003 2      FILL_FRAME (.SCB,
2074 3004 2          .SCB[PSM$B_FILE_BURST_CHAR],
2075 3005 2          PAGE_REF[0,.BOTTOM_OFFSET,.PAGE_WIDTH], .PAGE_WIDTH, 3);
2076 3006 2
2077 3007 2
2078 3008 2      FILL_FRAME (.SCB,
2079 3009 2          PAGE_REF[10,.BOTTOM_OFFSET,.PAGE_WIDTH], .PAGE_WIDTH-20, 3);
2080 3010 2
2081 3011 2      FILL_FRAME (.SCB,           ! Offset set... Add the diff
2082 3012 2          .SCB[PSM$B_JOB_BURST_CHAR],
2083 3013 2          PAGE_REF[14,.BOTTOM_OFFSET,.PAGE_WIDTH], .PAGE_WIDTH-28, 3);
2084 3014 2
2085 3015 2      ! re-init
2086 3016 2      STRING_DESC[SIZE] = K_MAX_BUFFER_SIZE;      ! reset buffer size
2087 3017 2
2088 3018 2      ! Get the sys$announce note and output to page
2089 3019 2      GET_DIGITAL_LOGO
2090 3020 2          (.SCB,
2091 3021 2          STRING_DESC[0],           ! SCB addr.
2092 3022 2          STRING_DESC[SIZE]);        ! Buffer descriptor
2093 3023 2
2094 3024 2      ! assume string will not over run the area... fail_safe is truncation

```

2094 3024 2 CENTER_FRAME (.SCB,
2095 3025 2 STRING_DESC[0],
2096 3026 2 PAGE_REF[0..BOTTOM_OFFSET+1..PAGE_WIDTH],
2097 3027 2 .PAGE_WIDTH, 1);
2098 3028 2
2099 3029 2
2100 3030 2 ! re-init
2101 3031 2 STRING_DESC[SIZE] = K_MAX_BUFFER_SIZE; ! reset buffer size
2102 3032 2
2103 3033 2 GET_RULER_COARSE
2104 3034 2 (.SCB,
2105 3035 2 STRING_DESC[0],
2106 3036 2 STRING_DESC[SIZE]); ! SCB addr.
2107 3037 2 ! Buffer descriptor
2108 3038 2 SCROLL_FRAME (.SCB,
2109 3039 2 STRING_DESC[0],
2110 3040 2 PAGE_REF[0..BOTTOM_OFFSET + 4..PAGE_WIDTH], .PAGE_WIDTH, 1);
2111 3041 2
2112 3042 2 SCROLL_FRAME (.SCB,
2113 3043 2 \$DESCRIPTOR ('1234567890'),
2114 3044 2 PAGE_REF[0..BOTTOM_OFFSET + 5..PAGE_WIDTH], .PAGE_WIDTH, 1);
2115 3045 2
2116 3046 2 ! Create a sentence describing the current job.
2117 3047 2
2118 3048 2 ! re-init
2119 3049 2 STRING_DESC[SIZE] = K_MAX_BUFFER_SIZE; ! reset buffer size
2120 3050 2
2121 3051 2 GET_JOB_DESCRIPTION
2122 3052 2 (.SCB,
2123 3053 2 0, ! SCB addr.
2124 3054 2 STRING_DESC[0], ! Use past tense
2125 3055 2 STRING_DESC[SIZE]); ! Buffer descriptor
2126 3056 2 ! Returned length
2127 3057 2 RET_LEN[0] = RETURN_FRAME_LENGTH
2128 3058 2 (.SCB,
2129 3059 2 STRING_DESC[0], ! string ref.
2130 3060 2 PAGE_REF[0,0..PAGE_WIDTH], ! ref to frame
2131 3061 2 .PAGE_WIDTH, ! cols to fill
2132 3062 2 .BOTTOM_OFFSET - .TOP_OFFSET); ! rows to fill
2133 3063 2
2134 3064 2 IF .RET_LEN[0] GTR 0
2135 3065 2 THEN
2136 3066 2 BOTTOM_OFFSET = .BOTTOM_OFFSET - (.RET_LEN[0] + 1);
2137 3067 2 ! adjust & allow for spacing
2138 3068 2 ! before inserting
2139 3069 2 ! insert the string delimited
2140 3070 2 INSERT_FRAME (.SCB,
2141 3071 2 STRING_DESC[0], ! string ref.
2142 3072 2 PAGE_REF[0..BOTTOM_OFFSET..PAGE_WIDTH], ! ref to frame
2143 3073 2 .PAGE_WIDTH, ! cols to fill
2144 3074 2 .RET_LEN[0]); ! rows to fill
2145 3075 2
2146 3076 2
2147 3077 2
2148 3078 2 ! Create a sentence describing the current file.
2149 3079 2
2150 3080 2 ! re-init

```
2151      3081 2     STRING_DESC[SIZE] = K_MAX_BUFFER_SIZE;      ! reset buffer size
2152      3082 2
2153      3083 2     GET_FILE_DESCRIPTION
2154          (.SCB,
2155          STRING_DESC[0],
2156          STRING_DESC[SIZE]);      ! SCB addr.
2157          ! Buffer descriptor
2158          ! Returned length
2159      3088 2     RET_LEN[0] = RETURN_FRAME_LENGTH
2160          (.SCB,
2161          STRING_DESC[0],
2162          PAGE_REF[0,0..PAGE_WIDTH],      ! string ref.
2163          :PAGE_WIDTH,      ! ref to frame
2164          :BOTTOM_OFFSET - .TOP_OFFSET);! cols to fill
2165          ! rows to fill
2166      3095 2     IF .RET_LEN[0] GTR 0
2167          THEN
2168              BOTTOM_OFFSET = .BOTTOM_OFFSET - (.RET_LEN[0] + 1);
2169          ! adjust & allow for spacing
2170          ! before inserting
2171          ! insert the string delimited
2172          INSERT_FRAME (.SCB,
2173              STRING_DESC[0],
2174              PAGE_REF[0..BOTTOM_OFFSET..PAGE_WIDTH],      ! string ref.
2175              ! ref to frame
2176              .PAGE_WIDTH,      ! cols to fill
2177              .RET_LEN[0]);      ! rows to fill
2178      3108 2     ! User name banner
2179      3109 2     !
2180      3110 2     RET_LEN[0] = INSERT_NAME_BANNER (
2181          .SCB,
2182          SCB_SIZE(USER NAME),      ! user name desc
2183          PAGE_REF[0..TOP_OFFSET..PAGE_WIDTH],
2184          ! ref to frame
2185          .PAGE_WIDTH,      ! max width Bann
2186          :BOTTOM_OFFSET - .TOP_OFFSET,
2187          ! frame size
2188          ?);      ! max hght Bann str
2189      3120 2     IF .RET_LEN[0] GTR 0
2190          THEN
2191              TOP_OFFSET = .TOP_OFFSET + .RET_LEN[0] + 2;
2192          ! adjust & allow for spacing
2193      3125 2     ! Get and insert the filename banner ... force the banner to be small (always)
2194      3126 2     !
2195      3127 2     ! re-init
2196      3128 2     STRING_DESC[SIZE] = K_MAX_BUFFER_SIZE;      ! reset buffer size
2197      3129 2
2198      3130 2     FORCE_LEN = 7;
2199      3131 2     IF .BOTTOM_OFFSET - .TOP_OFFSET LSS .FORCE_LEN
2200      3132 2     THEN
2201          FORCE_LEN = .BOTTOM_OFFSET - .TOP_OFFSET;
2202      3133 2
2203      3134 2
2204      3135 2     RET_LEN[0] = INSERT_FILENAME_BANNER
2205          (.SCB,
2206          STRING_DESC[0],      ! file name size
2207      3137 2
```

```

2208      3138 2          PAGE_REF[0..TOP_OFFSET..PAGE_WIDTH],
2209      3139 2          | ref to frame
2210      3140 2          .PAGE_WIDTH,
2211      3141 2          | max width Bann
2212      3142 2          :FORCE_LEN);
2213      3143 2          | max hght Bann str
2214      3144 2          IF .RET_LEN[0] GTR 0
2215      3145 2          THEN    TOP_OFFSET = .TOP_OFFSET + .RET_LEN[0] + 2;
2216      3146 2          | adjust & allow for spacing
2217      3147 2          ! Create a phrase which includes all the appropriate qualifiers
2218      3148 2          ! describing the current print and insert from the bottom without spacing.
2219      3149 2          ! re-init
2220      3150 2          STRING_DESC[SIZE] = K_MAX_BUFFER_SIZE;      ! reset buffer size
2221      3151 2          ! GET_QUALIFIERS
2222      3152 2          (.SCB,
2223      3153 2          STRING_DESC[0],           ! SCB addr.
2224      3154 2          STRING_DESC[SIZE]);        ! Buffer descriptor
2225      3155 2          | Returned length
2226      3156 2          RET_LEN[0] = RETURN_FRAME_LENGTH
2227      3157 2          (.SCB,
2228      3158 2          STRING_DESC[0],
2229      3159 2          PAGE_REF[0..PAGE_WIDTH],       ! string ref.
2230      3160 2          .PAGE_WIDTH-12,           ! ref to frame
2231      3161 2          .BOTTOM_OFFSET - .TOP_OFFSET); ! less twelve chars.
2232      3162 2          | rows to fill
2233      3163 2          ! IF .RET_LEN[0] GTR 0
2234      3164 2          THEN
2235      3165 2          BEGIN
2236      3166 2          BOTTOM_OFFSET = .BOTTOM_OFFSET - (.RET_LEN[0] + 1);
2237      3167 2          | adjust & allow for spacing
2238      3168 2          ! before inserting
2239      3169 2          ! move the string undelimited
2240      3170 2          MOVE_FRAME (.SCB,
2241      3171 2          $DESCRIPTOR ('Qualifiers: '),
2242      3172 2          PAGE_REF[0..BOTTOM_OFFSET..PAGE_WIDTH],
2243      3173 2          .PAGE_WIDTH,           ! ref to frame
2244      3174 2          :RETLEN[0]);         ! cols to fill
2245      3175 2          ! END;
2246      3176 2          ! insert the string delimited
2247      3177 2          INSERT_FRAME (.SCB,
2248      3178 2          STRING_DESC[0],
2249      3179 2          PAGE_REF[12..BOTTOM_OFFSET..PAGE_WIDTH], ! string ref.
2250      3180 2          .PAGE_WIDTH-12,           ! ref to frame
2251      3181 2          :RETLEN[0]);         ! cols to fill
2252      3182 2          ! rows to fill
2253      3183 2          ! 1 END;
2254      3184 2
2255      3185 2
2256      3186 2
2257      3187 2
2258      3188 2
2259      3189 2
2260      3190 2

```

SEPARATE
V04-001

Print Symbiont -- separation routines
FILL_FILE_TRAILER - Insert Information

1 2
16-Sep-1984 02:23:03
14-Sep-1984 22:32:26

VAX-11 Bliss-32 V4.0-742
[PRTSMB.SRC]SEPARATE.B32:2

Page 69
(18)

SEP
V04

20	3A	73	72	65	69	66	69	6C	61	75	51	0000000A,	00D8C	P.AAG:	.LONG	10
												00000000,	00D90		:ADDRESS	P.AAH
												0000000C	00D94	P.AAJ:	.ASCII	\Qualifiers: \
												00000000,	00D90	P.AAI:	.LONG	12
												00000000,	00D94		:ADDRESS	P.AAJ

OFFC 00000 FILL_FILE_TRAILER:

5B	0000V	CF	9E	00002	.WORD	Save R2, R3, R4, R5, R6, R7, R8, R9, R10, R11			2933
5A	0000V	CF	9E	00007	MOVAB	INSERT_FRAME, R1			
59	0000V	CF	9E	0000C	MOVAB	RETURN_FRAME LENGTH, R10			
5E	FDFC	CE	9E	00011	MOVAB	FILL FRAME, R9			
7E	0200	8F	3C	00016	MOVZWL	-5167(SP), SP			
04	AE	08	AE	9E 00018	MOVAB	#512, STRING_DESC			2953
			S2	D4 00020	CLRL	BUFFER, STRING_DESC+4			2954
53		10	AC	DO 00022	MOVL	TOP_OFFSET			2959
			03	DD 00026	PUSHL	PAGE_LENGTH, BOTTOM_OFFSET			2960
56		0C	AC	DO 00028	MOVL	PAGE_WIDTH, R6			2964
			56	DD 0002C	PUSHL	R6			
54		08	AC	DO 0002E	MOVL	PAGE_REF, R4			
55		02	A2	9E 00032	MOVAB	2(R2), R5			
55			56	C4 00036	MULL2	R6, R5			
			6544	9F 00039	PUSHAB	(R5)[R4]			
58		04	AC	DO 0003C	MOVL	SCB, R8			2963
7E	02A4	C8	9A	00040	MOVZBL	676(R8), -(SP)			2964
			58	DD 00045	PUSHL	R8			
69			05	FB 00047	CALLS	#5, FILL_FRAME			2968
			03	DD 0004A	PUSHL	#3			
		EC	A6	9F 0004C	PUSHAB	-20(R6)			
		0A	A544	9F 0004F	PUSHAB	10(R5)[R4]			
			20	DD 00053	PUSHL	#32			
			58	DD 00055	PUSHL	R8			
69			05	FB 00057	CALLS	#5, FILL_FRAME			2972
			03	DD 0005A	PUSHL	#3			
		E6	A6	9F 0005C	PUSHAB	-26(R6)			
		0D	A544	9F 0005F	PUSHAB	13(R5)[R4]			
7E	02A6	C8	9A	00063	MOVZBL	678(R8), -(SP)			
			58	DD 00068	PUSHL	R8			
69			05	FB 0006A	CALLS	#5, FILL_FRAME			2975
6E	0200	8F	3C	0006D	MOVZWL	#512, STRING_DESC			2979
			5E	DD 00072	PUSHL	SP			2978
		04	AE	9F 00074	PUSHAB	STRING_DESC			2977
			58	DD 00077	PUSHL	R8			
0000V	CF		03	FB 00079	CALLS	#3, GET_EOF			2984
			07	DD 0007E	PUSHL	#7			
7E	53		52	C3 00080	SUBL3	TOP_OFFSET, BOTTOM_OFFSET, -(SP)			2987
55			56	DD 00084	PUSHL	R6			2986
		52	56	C5 00086	MULL3	R6, TOP_OFFSET, R5			2984
			6544	9F 0008A	PUSHAB	(R5)[R4]			
		10	AE	9F 0008D	PUSHAB	STRING_DESC			2983
			58	DD 00090	PUSHL	R8			2984
0000V	CF		06	FB 00092	CALLS	#6, INSERT_NAME_BANNER			2992
	57		50	DO 00097	MOVL	R0, RET_LEN			
			05	15 0009A	BLEQ	18			2994
		52	A742	9E 0009C	MOVAB	2(RET_LEN)[TOP_OFFSET], TOP_OFFSET			

SEPARATE
V04-001Print Symbiont -- separation routines
FILL_FILE_TRAILER - Insert Information into theJ 2
16-Sep-1984 02:23:03
14-Sep-1984 22:32:26VAX-11 Bliss-32 V4.0-742
[PRTSMB.SRC]SEPARATE.B32;2Page 70
(18)SEP
V04

53	10	AC	05	C3 000A1	1\$:	SUBL3	#5, PAGE_LENGTH, BOTTOM_OFFSET	3000
			03	DD 000A6		PUSHL	#3	3004
55	53		56	DD 000A8		PUSHL	R6	
			56	C5 000AA		MULL3	R6, BOTTOM_OFFSET, R5	
		02A4	6544	9F 000AE		PUSHAB	(R5)[R4]	
	7E		C8	9A 000B1		MOVZBL	676(R8), -(SP)	
			58	DD 000B6		PUSHL	R8	
	69		05	FB 000B8		CALLS	#5, FILL_FRAME	
			03	DD 000BB		PUSHL	#3	
			EC	A6 9F 000BD		PUSHAB	-20(R6)	
			0A	A544 9F 000C0		PUSHAB	10(R5)[R4]	
			20	DD 000C4		PUSHL	#32	
	69		58	DD 000C6		PUSHL	R8	
			05	FB 000C8		CALLS	#5, FILL_FRAME	
			03	DD 000CB		PUSHL	#3	
			E4	A6 9F 000CD		PUSHAB	-28(R6)	
			0E	A544 9F 000D0		PUSHAB	14(R5)[R4]	
	7E	02A6	C8	9A 000D4		MOVZBL	678(R8), -(SP)	
			58	DD 000D9		PUSHL	R8	
	69		05	FB 000DB		CALLS	#5, FILL_FRAME	
			6E	0200 8F 3C 000DE		MOVZWL	#512, STRING_DESC	
			04	SE DD 000E3		PUSHL	SP	
			AE	9F 000E5		PUSHAB	STRING_DESC	
			58	DD 000E8		PUSHL	R8	
0000V	CF		03	FB 000EA		CALLS	#3, GET_DIGITAL_LOGO	
			01	DD 000EF		PUSHL	#1	
	55		56	DD 000F1		PUSHL	R6	
	55		01	A3 9E 000F3		MOVAB	1(R3), R5	
			56	C4 000F7		MULL2	R6, R5	
			6544	9F 000FA		PUSHAB	(R5)[R4]	
			OC	AE 9F 000FD		PUSHAB	STRING_DESC	
			58	DD 00100		PUSHL	R8	
0000V	CF		05	FB 00102		CALLS	#5, CENTER_FRAME	
	6E	0200	8F	3C 00107		MOVZWL	#512, STRING_DESC	
			04	5E DD 0010C		PUSHL	SP	
			AE	9F 0010E		PUSHAB	STRING_DESC	
0000V	CF		58	DD 00111		PUSHL	R8	
			03	FB 00113		CALLS	#3, GET_RULER_COARSE	
			01	DD 00118		PUSHL	#1	
	55		56	DD 0011A		PUSHL	R6	
	55		04	A3 9E 0011C		MOVAB	4(R3), R5	
			56	C4 00120		MULL2	R6, R5	
			6544	9F 00123		PUSHAB	(R5)[R4]	
			OC	AE 9F 00126		PUSHAB	STRING_DESC	
0000V	CF		58	DD 00129		PUSHL	R8	
			05	FB 0012B		CALLS	#5, SCROLL_FRAME	
			01	DD 00130		PUSHL	#1	
	55		56	DD 00132		PUSHL	R6	
	55		05	A3 9E 00134		MOVAB	5(R3), R5	
			56	C4 00138		MULL2	R6, R5	
			6544	9F 0013B		PUSHAB	(R5)[R4]	
		FEA2	CF	9F 0013E		PUSHAB	P.AAG	
0000V	CF		58	DD 00142		PUSHL	R8	
	6E	0200	05	FB 00144		CALLS	#5, SCROLL_FRAME	
			8F	3C 00149		MOVZWL	#512, STRING_DESC	
			SE	DD 0014E		PUSHL	SP	
			AE	9F 00150		PUSHAB	STRING_DESC	

SEPARATE
V04-001

Print Symbiont -- separation routines
FILL-FILE-TRAILER - Insert Information

K 2
16-Sep-1984 02:23:03
14-Sep-1984 22:32:26

VAX-11 Bliss-32 V4.0-742
[PRTSMB.SRC]SEPARATE.B32;2

Page 71
(18)

SEP
V04

7E	0000V	CF	7E	D4	00153	CLRL	-(SP)
		53	58	DD	00155	PUSHL	R8
			04	FB	00157	CALLS	#4, GET_JOB_DESCRIPTION
		0050	52	C3	0015C	SUBL3	TOP_OFFSET, BOTTOM_OFFSET, -(SP)
		OC	8F	BB	00160	PUSHR	#^MZR4,R6>
			AE	9F	00164	PUSHAB	STRING_DESC
			58	DD	00167	PUSHL	R8
50		6A	05	FB	00169	CALLS	#5, RETURN_FRAME_LENGTH
		57	50	DO	0016C	MOVL	R0, RET_LEN
50		53	08	15	0016F	BLEQ	2\$
		53	57	C3	00171	SUBL3	RET_LEN, BOTTOM_OFFSET, R0
		FF	A0	9E	00175	MOVAB	-1(R0), BOTTOM_OFFSET
50		7E	56	7D	00179	MOVQ	R6, -(SP)
		53	56	C5	0017C	MULL3	R6, BOTTOM_OFFSET, R0
			6044	9F	00180	PUSHAB	(R0)[R4]
		OC	AE	9F	00183	PUSHAB	STRING_DESC
			58	DD	00186	PUSHL	R8
6B		6E	05	FB	00188	CALLS	#5, INSERT_FRAME
		0200	8F	3C	0018B	MOVZWL	#512, STRING_DESC
			5E	DD	00190	PUSHL	SP
			04	AE	9F	PUSHAB	STRING_DESC
			58	DD	00195	PUSHL	R8
7E	0000V	CF	03	FB	00197	CALLS	#3, GET_FILE_DESCRIPTION
		53	52	C3	0019C	SUBL3	TOP_OFFSET, BOTTOM_OFFSET, -(SP)
			8F	BB	001A0	PUSHR	#^MZR4,R6>
		0050	AE	9F	001A4	PUSHAB	STRING_DESC
		OC	58	DD	001A7	PUSHL	R8
			58	FB	001A9	CALLS	#5, RETURN_FRAME_LENGTH
50		6A	50	DO	001AC	MOVL	R0, RET_LEN
		57	08	15	001AF	BLEQ	3\$
50		53	57	C3	001B1	SUBL3	RET_LEN, BOTTOM_OFFSET, R0
		53	A0	9E	001B5	MOVAB	-1(R0), BOTTOM_OFFSET
50		7E	56	7D	001B9	MOVQ	R6, -(SP)
		53	56	C5	001BC	MULL3	R6, BOTTOM_OFFSET, R0
			6044	9F	001C0	PUSHAB	(R0)[R4]
		OC	AE	9F	001C3	PUSHAB	STRING_DESC
			58	DD	001C6	PUSHL	R8
6B			05	FB	001C8	CALLS	#5, INSERT_FRAME
			07	DD	001CB	PUSHL	#7
7E		53	52	C3	001CD	SUBL3	TOP_OFFSET, BOTTOM_OFFSET, -(SP)
			56	DD	001D1	PUSHL	R6
55		52	56	C5	001D3	MULL3	TOP_OFFSET, R5
			6544	9F	001D7	PUSHAB	(R5)[R4]
		016C	C8	9F	001DA	PUSHAB	364(R8)
			58	DD	001DE	PUSHL	R8
0000V		CF	06	FB	001E0	CALLS	#6, INSERT_NAME_BANNER
		57	50	DO	001E5	MOVL	R0, RET_LEN
			05	15	001E8	BLEQ	4\$
50		52	02	A742	9E	MOVAB	2(RET_LEN)[TOP_OFFSET], TOP_OFFSET
		6E	0200	8F	3C	MOVZWL	#512, STRING_DESC
			07	DO	001EF	MOVQ	#7, FORCE_LEN
50		53	52	C3	001F7	SUBL3	TOP_OFFSET, BOTTOM_OFFSET, R0
		51	50	D1	001FB	CMPL	R0, FORCE_LEN
			03	18	001FE	BGEQ	5\$
		51	50	DO	00200	MOVL	R0, FORCE_LEN
			51	DD	00203	PUSHL	FORCE_LEN
		56	56	DD	00205	PUSHL	R6

SEPARATE
V04-001Print Symbiont -- separation routines
FILL_FILE_TRAILER - Insert Information into theL 2
16-Sep-1984 02:23:03
14-Sep-1984 22:32:26VAX-11 Bliss-32 V4.0-742
[PRTSMB.SRC]SEPARATE.B32;2Page 72
(18)SEP
V04

55	52		S6	C5	00207	MULL3	R6, TOP_OFFSET, R5	: 3138
		0C	6544	9F	0020B	PUSHAB	(R5)[R4]	: 3137
			AE	9F	0020E	PUSHAB	STRING_DESC	: 3138
			58	DD	00211	PUSHL	R8	: 3138
	0000V	CF	05	FB	00213	CALLS	#5, INSERT_FILENAME_BANNER	: 3143
		57	50	DO	00218	MOVL	R0, RET_LEN	: 3145
			05	15	0021B	BLEQ	6\$: 3145
		52	02	A742	9E	MOVAB	2(RET_LEN)[TOP_OFFSET], TOP_OFFSET	: 3145
		6E	0200	8F	3C	MOVZWL	#512, STRING_DESC	: 3152
			5E	DD	00227	PUSHL	SP	: 3157
			04	AE	00229	PUSHAB	STRING_DESC	: 3156
			58	DD	0022C	PUSHL	R8	: 3155
7E	0000V	CF	03	FB	0022E	CALLS	#3, GET_QUALIFIERS	: 3164
		53	52	C3	00233	SUBL3	TOP_OFFSET, BOTTOM_OFFSET, -(SP)	: 3163
			F4	A6	9F	PUSHAB	-12(R6)	: 3162
			54	DD	0023A	PUSHL	R4	: 3161
			0C	AE	0023C	PUSHAB	STRING_DESC	: 3162
			58	DD	0023F	PUSHL	R8	: 3162
		6A	05	FB	00241	CALLS	#5, RETURN_FRAME_LENGTH	: 3166
		57	50	DO	00244	MOVL	R0, RET_LEN	: 3169
			1D	15	00247	BLEQ	7\$: 3166
50	53	57	57	C3	00249	SUBL3	RET_LEN, BOTTOM_OFFSET, R0	: 3169
	53	FF	A0	9E	0024D	MOVAB	-1(R0), BOTTOM_OFFSET	: 3178
	7E		56	7D	00251	MOVA	R6, -(SP)	: 3176
50	53		56	C5	00254	MULL3	R6, BOTTOM_OFFSET, R0	: 3176
			6044	9F	00258	PUSHAB	(R0)[R4]	: 3175
		FD99	CF	9F	0025B	PUSHAB	P.AAI	: 3176
			58	DD	0025F	PUSHL	R8	: 3176
	0000V	CF	05	FB	00261	CALLS	#5, MOVE_FRAME	: 3188
			57	DD	00266	PUSHL	RET LEN	: 3187
		53	F4	A6	9F	PUSHAB	-12(R6)	: 3187
			56	C4	0026B	MULL2	R6, R3	: 3185
			0C	A344	9F	PUSHAB	12(R3)[R4]	: 3184
			OC	AE	00272	PUSHAB	STRING_DESC	: 3185
		6B	58	DD	00275	PUSHL	R8	: 3185
			05	FB	00277	CALLS	#5, INSERT_FRAME	: 3190
			04	0027A		RET		

: Routine Size: 635 bytes, Routine Base: CODE + 0DA8

```

2262 3191 1 %sbttl 'RETURN_FRAME_LENGTH - Returns the Frame Length for String Insertion'
2263 3192 1 ++
2264 3193 1 Functional Description:
2265 3194 1 Returns the frame length needed to insert the string into the page.
2266 3195 1 This routine checks the top of frame/bottom_of_frame offsets and
2267 3196 1 decides if the string will fit=
2268 3197 1 1) yes - return number of frame rows required for string to fit.
2269 3198 1 2) no - return zero
2270 3199 1
2271 3200 1
2272 3201 1 Formal Parameters:
2273 3202 1 SCB - Address of the SCB
2274 3203 1 STR_DESC - Descriptor of String to Insert
2275 3204 1 FRAME_PTR - Address of first byte of Frame
2276 3205 1 FRAME_LENGTH - Length of Frame
2277 3206 1 FRAME_WIDTH - Width of Frame
2278 3207 1
2279 3208 1
2280 3209 1 Implicit Inputs:
2281 3210 1 none
2282 3211 1
2283 3212 1 Implicit Outputs:
2284 3213 1 none
2285 3214 1
2286 3215 1 Returned Value:
2287 3216 1 none
2288 3217 1
2289 3218 1 Side Effects:
2290 3219 1 none
2291 3220 1 --
2292 3221 1 ROUTINE RETURN_FRAME_LENGTH (
2293 3222 1 SCB : REF SBBLOCK,
2294 3223 1 STR_DESC : REF VECTOR[2],
2295 3224 1 FRAME_PTR : REF PAGE_ARRAY,
2296 3225 1 FRAME_WIDTH . . . ! Number of Columns
2297 3226 1 FRAME_LENGTH . . . ! Number of Rows
2298 3227 1 ) =
2299 3228 2 BEGIN
2300 3229 2
2301 3230 2 LOCAL MAX_CHARS;
2302 3231 2
2303 3232 2 ! don't even try if there is no frame left
2304 3233 2 IF (.FRAME_LENGTH LEQ 0) OR
2305 3234 2 (.FRAME_WIDTH LEQ 0) OR
2306 3235 2 (.STR_DESC[SIZE] EQL 0)
2307 3236 2 THEN
2308 3237 2 RETURN 0;
2309 3238 2
2310 3239 2 MAX_CHARS = .FRAME_WIDTH * .FRAME_LENGTH;
2311 3240 2
2312 3241 2 ! The boundary condition of string size of some multiple of frame width
2313 3242 2 can occur - add one less than the frame width to overcome this condition
2314 3243 2
2315 3244 2 IF .STR_DESC[SIZE] LEQ .MAX_CHARS
2316 3245 2 THEN
2317 3246 2 RETURN ((.STR_DESC[SIZE]+(.FRAME_WIDTH-1)) / .FRAME_WIDTH);
2318 3247 2

```

```
: 2319 3248 2 ! otherwise it just won't fit
: 2320 3249 2 RETURN 0;
: 2321 3250 2
: 2322 3251 1 END:
```

0000 00000 RETURN_FRAME LENGTH:

				.WORD	Save nothing	
			14 AC D5 00002	TSTL	FRAME_LENGTH	: 3221
			23 15 00005	BLEQ	1\$: 3233
			10 AC D5 00007	TSTL	FRAME_WIDTH	: 3234
			1E 15 0000A	BLEQ	1\$: 3235
			08 BC D5 0000C	TSTL	ASTR_DESC	: 3239
			19 13 0000F	BEQL	1\$: 3244
50	10	AC	14 AC C5 00011	MULL3	FRAME_LENGTH, FRAME_WIDTH, MAX_CHARS	: 3246
			50 08 BC D1 00017	CMPL	ASTR_DESC, MAX_CHARS	: 3251
50	08	BC	10 AC C1 0001D	BGTR	1\$	
			50 50 D7 00023	ADDL3	FRAME_WIDTH, ASTR_DESC, R0	
			10 AC C6 00025	DECL	R0	
			04 00029	DIVL2	FRAME_WIDTH, R0	
			50 D4 0002A 1\$:	RET		
			04 0002C	CLRL	R0	

: Routine Size: 45 bytes. Routine Base: CODE + 1023

```
2324 3252 1 %sbttl 'RETURN_FRAME_WIDTH - Returns the Frame Length for String Insertion'
2325 3253 1 ++
2326 3254 1 Functional Description:
2327 3255 1 Returns the frame width needed to insert the required lengths into
2328 3256 1 the page. This routine returns only a prescribed value and
2329 3257 1 decides if the string will fit -
2330 3258 1 Return value - frame width
2331 3259 1 Return zero - only if no length or width of frame.
2332 3260 1 Assumes the FRAME_LENGTH is constant. (How many lengths are needed
2333 3261 1 to fit this string)
2334 3262 1
2335 3263 1 Formal Parameters:
2336 3264 1 SCB      - Address of the SCB
2337 3265 1 STR_DESC - Descriptor of String to Insert
2338 3266 1 FRAME_PTR - Address of first byte of Frame
2339 3267 1 FRAME_LENGTH - Length of Frame
2340 3268 1 FRAME_WIDTH - Width of Frame
2341 3269 1
2342 3270 1
2343 3271 1 Implicit Inputs:
2344 3272 1      none
2345 3273 1
2346 3274 1 Implicit Outputs:
2347 3275 1      none
2348 3276 1
2349 3277 1 Returned Value:
2350 3278 1      none
2351 3279 1
2352 3280 1 Side Effects:
2353 3281 1      none
2354 3282 1 ---
2355 3283 1 ROUTINE RETURN_FRAME_WIDTH (
2356 3284 1      SCB      : REF $BBLOCK
2357 3285 1      STR_DESC : REF VECTORE[2]
2358 3286 1      FRAME_PTR : REF PAGE_ARRAY
2359 3287 1      FRAME_WIDTH .
2360 3288 1      FRAME_LENGTH .          ! Number of Columns
2361 3289 1      ) =           ! Number of Rows
2362 3290 2 BEGIN
2363 3291 2
2364 3292 2 LOCAL
2365 3293 2      MAX_CHARS:
2366 3294 2      TEMP_WIDE:
2367 3295 2
2368 3296 2      ! don't even try if there is no frame left
2369 3297 2      IF (.FRAME_LENGTH LEQ 0) OR
2370 3298 2      (.FRAME_WIDTH LEQ 0) OR
2371 3299 2      (.STR_DESC[SIZE] EQL 0)
2372 3300 2 THEN
2373 3301 2      RETURN 0;
2374 3302 2
2375 3303 2      MAX_CHARS = .FRAME_WIDTH * .FRAME_LENGTH;
2376 3304 2      IF _MAX_CHARS LSS .STR_DESC[SIZE]
2377 3305 2 THEN
2378 3306 2      RETURN 0;                      ! string wont fit
2379 3307 2
2380 3308 2      ! The boundary condition of string size of some multiple of frame width
```

SEPARATE
V04-001

Print Symbiont -- separation routines
RETURN_FRAME_WIDTH - Returns the Frame Length

C 3
16-Sep-1984 02:23:03
14-Sep-1984 22:32:26

VAX-11 Bliss-32 V4.0-742
[PRTSMB.SRC]SEPARATE.B32;2

Page 76
(20)

```
: 2381 3309 2 ! can occur - add one less than the frame width to overcome this condition
: 2382 3310 2 !
: 2383 3311 2 TEMP_WIDE = .STR_DESC[SIZE]/.FRAME_LENGTH;
: 2384 3312 2
: 2385 3313 2 IF .TEMP_WIDE LEQ .FRAME_WIDTH
: 2386 3314 2 THEN
: 2387 3315 2     RETURN (.TEMP_WIDE);           ! return the value
: 2388 3316 2
: 2389 3317 2 ! otherwise it just won't fit
: 2390 3318 2 RETURN 0;
: 2391 3319 2
: 2392 3320 1 END;
```

0000 00000 RETURN_FRAME_WIDTH:							
					.WORD	Save nothing	3283
		51	14	AC	D0	00002	3297
				20	15	00006	
			10	AC	D5	00008	3298
				1B	15	0000B	
			08	BC	D5	0000D	3299
				16	13	00010	
50	10	AC		51	C5	00012	3303
	08	BC		50	D1	00017	3304
50	08	BC		08	19	0001B	
	10	AC		51	C7	0001D	3311
				50	D1	00022	3313
				02	15	00026	
				50	D4	00028 1\$:	3320
				04	0002A 2\$:		
					CLRL	R0	
					RET		

; Routine Size: 43 bytes, Routine Base: CODE + 1050

SEP
V04

2394 3321 1 %sbttl 'GET_REVISION_DATE - Get the revision date of current file'
2395 3322 1 !++
2396 3323 1 | Functional Description:
2397 3324 1 | This routine creates a phrase with DD-MMM-YYYY HH:MM describing
2398 3325 1 | the revision date of the current file. Returns zero if file
2399 3326 1 | not open.
2400 3327 1 |
2401 3328 1 | Formal Parameters:
2402 3329 1 | SCB Address of the SCB
2403 3330 1 | STR_DESC Desc of String to Return
2404 3331 1 | RET_LEN Return Length of Desc.
2405 3332 1 |
2406 3333 1 | Implicit Inputs:
2407 3334 1 | none
2408 3335 1 |
2409 3336 1 | Implicit Outputs:
2410 3337 1 | none
2411 3338 1 |
2412 3339 1 | Returned Value:
2413 3340 1 | none
2414 3341 1 |
2415 3342 1 | Side Effects:
2416 3343 1 | none
2417 3344 1 |--
2418 3345 1 ROUTINE GET_REVISION_DATE (SCB : REF \$BBLOCK,
2419 3346 1 : REF VECTOR[,2], ! SCB
2420 3347 1 : REF VECTOR [,WORD] ! Output buffer desc
2421 3348 1 : NOVALUE = ! Return length (word)
2422 3349 1)
2423 3350 2 BEGIN
2424 3351 2 BIND
2425 3352 2 XABDAT = .SCB[PSMSA_XABDAT]: \$BBLOCK, ! - RMS date block
2426 3353 2 FORMAT = \$DESCRIPTOR (!
2427 3354 2 '!17%D'), ! - revision date
2428 3355 2
2429 3356 2 NAM = .SCB[PSMSA_NAM]: REF \$BBLOCK;
2430 3357 2
2431 3358 2 LOCAL CURRENT_LEN : INITIAL (0);
2432 3359 2
2433 3360 2
2434 3361 2 IF FILE_OPEN(.SCB)
2435 3362 2 THEN
P 3363 2 SFAO (!
P 3364 2 FORMAT,
P 3365 2 CURRENT_LEN,
P 3366 2 STR_DESC[0],
P 3367 2 XABDAT[XAB\$0_RDT],
P 3368 2);
2441 3369 2
2442 3370 2 RET_LEN[0] = .CURRENT_LEN;
2443 3371 2
2444 3372 1 END;

44 25 37 31 21 0107B P.AAL: .ASCII \!17%D\
00000005 01080 P.AAK: .LONG 5

SEPARATE
V04-001

Print Symbiont -- separation routines
GET_REVISION_DATE - Get the revision date of cu

E 3
16-Sep-1984 02:23:03
14-Sep-1984 22:32:26

VAX-11 Bliss-32 V4.0-742
[PRTSMB.SRC]SEPARATE.B32;2

Page 78
(21)

SEI
VOI

00000000' 01084 .ADDRESS P.AAL

FORMAT= P.AAK

0004 00000 GET_REVISION_DATE:										
										.WORD Save R2
50	52	04	AC	DD	00002		MOVL	SCB, R0		: 3345
		0254	CO	DD	00006		MOVL	596(R0), R2		: 3352
			7E	D4	0000B		CLRL	CURRENT_LEN		: 3356
			50	DD	0000D		PUSHL	R0		: 3361
0000V	CF		01	FB	0000F		CALLS	#1, FILE_OPEN		
	13		50	E9	00014		BLBC	R0, 1\$		
		OC	A2	9F	00017		PUSHAB	12(R2)		3368
		08	AC	DD	0001A		PUSHL	STR DESC		
		08	AE	9F	0001D		PUSHAB	CURRENT_LEN		
00000000G	00	D5	AF	9F	00020		PUSHAB	FORMAT		21
OC	BC		04	FB	00023	1\$:	CALLS	#4, SYSSFAO		
			6E	B0	0002A		MOVW	CURRENT_LEN, RET_LEN		4C
			04	0002E			RET			3370
										3372

: Routine Size: 47 bytes. Routine Base: CODE + 1088

21
20

```

2447 3373 1 %sbttl 'GET_SYSTEM_ANNOUNCEMENT- Create a Sentence Describing the Current Job'
2448 3374 1 ++
2449 3375 1 Functional Description:
2450 3376 1 This routine get the system annoucement. All allocation of buffers
2451 3377 1 handled by caller
2452 3378 1
2453 3379 1 Formal Parameters:
2454 3380 1 SCB - Address of the SCB
2455 3381 1 STR_DESC - Desc of String to Return
2456 3382 1 RET_LEN - Return length of Desc.
2457 3383 1
2458 3384 1 Implicit Inputs:
2459 3385 1 none
2460 3386 1
2461 3387 1 Implicit Outputs:
2462 3388 1 none
2463 3389 1
2464 3390 1 Returned Value:
2465 3391 1 none
2466 3392 1
2467 3393 1 Side Effects:
2468 3394 1 none
2469 3395 1 --
2470 3396 1 ROUTINE GET_SYSTEM_ANNOUNCEMENT (
2471 3397 1 SCB : REF $BLOCK, ! SCB
2472 3398 1 STR_DESC : REF VECTOR[2], ! Output buffer desc
2473 3399 1 RSL_LEN : REF VECTOR [,WORD] ! Return length (word)
2474 3400 1 ) : NOVALUE =
2475 3401 2 BEGIN
2476 3402 2 BIND
2477 P 3403 2 DEFINED_ANNOUNCE = $DESCRIPTOR (
2478 3404 2 '!' AF' ), ! - Defined announcement
2479 3405 2
2480 P 3406 2 ANNOUNCE = $DESCRIPTOR (
2481 3407 2 'PMS$ANNOUNCE' ); ! - system annoucement
2482 3408 2
2483 3409 2 LOCAL
2484 3410 2 FAO_DESC : VECTOR[2],
2485 3411 2 BUFFER : VECTOR[256,byte],
2486 3412 2 TEMP_LEN : INITIAL (0),
2487 3413 2 STATUS :
2488 3414 2
2489 3415 2 FAO_DESC[SIZE] = %ALLOCATION(BUFFER);
2490 3416 2 FAO_DESC[ADDR] = BUFFER;
2491 3417 2
2492 3418 2 ! STATUS = STRNLNM(attr =LNMSM_CASE BLIND,
2493 3419 2 tabnam=%ASCID-'LNMSYSTEM_TABLE',
2494 3420 2 lognam= ANNOUNCE,
2495 3421 2 rsllen= FAO_DESC[SIZE],
2496 3422 2 rslbuf= FAO_DESC[ADDR];
2497 3423 2
2498 P 3424 2 STATUS = $TRNLOG(lognam= ANNOUNCE,
2499 3425 2 rsbuf= FAO_DESC,
2500 3426 2 rsllen= TEMP_LEN);
2501 3427 2
2502 3428 2 IF .STATUS
2503 3429 2 THEN ! Success - Normal, Buffer_overflow

```

```

2504 3430 2
2505 3431 2 | check for command file pointer "a" sign or no-translation code
2506 3432 3
2507 3433 3 BEGIN
2508 3434 4 IF (.STATUS EQL SSS_NOTTRAN)
2509 3435 3 THEN
2510 3436 3 TEMP_LEN = 0;
2511 3437 3 END
2512 3438 2 ELSE ! Bad status - Badparam, Badlength,
2513 3439 2 TEMP_LEN = 0; ! Badtable, Notfound(badname), Baddepth
2514 3440 2
2515 3441 2 IF .TEMP_LEN EQL 0 ! Get the default sys version #
2516 3442 2 THEN
2517 3443 2 ! Put in the Digital Logo
2518 3444 2 GET_DIGITAL_LOGO(.SCB,STR_DESC[0],RSL_LEN[0])
2519 3445 2 ELSE
2520 P 3446 2 SFAO (
2521 P 3447 2 DEFINED_ANNOUNCE,
2522 P 3448 2 RSL_LEN[0],
2523 P 3449 2 STR_DESC[0],
2524 P 3450 2 .TEMP_LEN,
2525 P 3451 2 .FAO_DESC[ADDR]);
2526 3452 2
2527 3453 1 END;

```

45 43 4E 55 4F 4E 4E 41 24 4D	46 41 21 010B7 P.AAN: .ASCII \!AF\
	010BA .BLKB 2
	00000003 010BC P.AAM: .LONG 3
	00000000 010C0 P.AAP: .ADDRESS P.AAN
	0000000C 010D0 P.AAO: .LONG 12
	00000000 010D4 .ADDRESS P.AAP

DEFINED_ANNOUNCE= P.AAM
 ANNOUNCE= P.AAO
 .EXTRN SYS\$TRNLOG

0000 00000 GET_SYSTEM_ANNOUNCEMENT:							
	SE	FEF8	CE 9E 00002		.WORD	Save nothing	: 3396
F8	AD	0100	7E D4 00007	MOVAB	-264(SP), SP		: 3401
FC	AD	04	8F 3C 00009	CLRL	TEMP_LEN		: 3415
			AE 9E 0000F	MOVZWL	#256, FAO_DESC		: 3416
			7E 7C 00014	MOVAB	BUFFER, FAO_DESC+4		: 3426
			7E D4 00016	CLRQ	-(SP)		
		F8	9F 00018	CLRL	-(SP)		
		AD	PUSHAB FAO_DESC				
		10	AE 9F 0001B	PUSHAB	TEMP_LEN		
		D7	AF 9F 0001E	PUSHAB	ANNOUNCE		
00000000G	00		06 FB 00021	CALLS	#6, SYS\$TRNLOG		
00000629	09		50 E9 00028	BLBC	STATUS, 18		: 3428
	8F		50 D1 0002B	CMPL	STATUS, #1577		: 3434
			02 12 00032	BNEQ	28		
			6E D4 00034	CLRL	TEMP_LEN		: 3439
			6E D5 00036	TSTL	TEMP_LEN		: 3441
			1\$: OD 12 00038	BNEQ	38		

SEPARATE
V04-001

Print Symbiont -- separation routines
GET_SYSTEM_ANNOUNCEMENT- Create a Sentence Desc

H 3
16-Sep-1984 02:23:03
14-Sep-1984 22:32:26

VAX-11 Bliss-32 V4.0-742
[PRTSMB.SRC]SEPARATE.B32;2

Page 81
(22)

7E	08	AC	7D	0003A	MOVQ	STR_DESC, -(SP)	: 3444
0000V CF	04	AC	DD	0003E	PUSHL	SCB	
	03	FB	00041		CALLS	#3, GET_DIGITAL_LOGO	
	04		00046		RET		
	FC	AD	DD	00047 38:	PUSHL	FAO_DESC+4	
	04	AE	DD	0004A	PUSHL	TEMP LEN	3451
	08	AC	DD	0004D	PUSHL	STR_DESC	
	0C	AC	DD	00050	PUSHL	RSL LEN	
00000000G 00	BE	AF	9F	00053	PUSHAB	DEFINED ANNOUNCE	
	05	FB	00056		CALLS	#5, SYSSFAO	
	04		0005D		RET		

; Routine Size: 94 bytes. Routine Base: CODE + 10D8

```
2529      3454 1 %sbttl 'GET_VMS_LOGO - Create a Phrase of VMS logo'  
2530      3455 1 !++  
2531      3456 1 | Functional Description:  
2532          VAX/VMS Version Vx.x  
2533      3458 1 |  
2534      3459 1 | Formal Parameters:  
2535          SCB           - Address of the SCB  
2536          STR_DESC      - Desc of String to Return  
2537          RET_LEN       - Return length of Desc.  
2538      3463 1 |  
2539      3464 1 | Implicit Inputs:  
2540          none  
2541      3466 1 |  
2542      3467 1 | Implicit Outputs:  
2543          none  
2544      3468 1 |  
2545      3469 1 | Returned Value:  
2546          none  
2547      3472 1 |  
2548      3473 1 | Side Effects:  
2549          none  
2550      3475 1 |--  
2551      3476 1 ROUTINE GET_VMS_LOGO (  
2552          SCB           : REF $BBLOCK,          ! SCB  
2553          STR_DESC      : REF VECTOR[2],        ! Output buffer desc  
2554          RSL_LEN       : REF VECTOR [,WORD]    ! Return length (word)  
2555          )           : NOVALUE =  
2556      3481 2 BEGIN  
2557      3482 2 BIND  
2558      3483 2 TRAILING = 1,  
2559      3484 2  
2560      P 3485 2 DEFAULT = $DESCRIPTOR (   
2561          'VAX/VMS'     .  
2562          'VAX/VMS'     .  
2563          'VAX/VMS'     .  
2564          'VAX/VMS'     .  
2565          'VAX/VMS'     .  
2566          'VAX/VMS'     .  
2567          'VAX/VMS'     .  
2568          'VAX/VMS'     .  
2569          'VAX/VMS'     .  
2570          'VAX/VMS'     .  
2571          'VAX/VMS'     .  
2572          'VAX/VMS'     .  
2573          'VAX/VMS'     .  
2574          'VAX/VMS'     .  
2575          'VAX/VMS'     .  
2576          'VAX/VMS'     .  
2577          'VAX/VMS'     .  
2578          'VAX/VMS'     .  
2579          'VAX/VMS'     .  
2580          'VAX/VMS'     .  
2581          'VAX/VMS'     .  
2582          'VAX/VMS'     .  
2583          'VAX/VMS'     .  
2584          'VAX/VMS'     .  
2585          'VAX/VMS'     );
```

```

2586 3511 2 LOCAL
2587 3512 3513 2 STR_PTR
2588 3514 3515 2 STR_LEN :
2589
2590
2591 3516 IF .SCB[PSMSL_PAGE_WIDTH] LSS 20
2592 3517 THEN ! no room for burst bar
2593 3518 BEGIN
2594 3519 RSL_LEN[0] = 0;
2595 3520 RETURN;
2596 3521 END:
2597
2598 P 3522 3523 $FA0 (
2599 P 3524 3525 2 DEFAULT
2600 3526 3527 2 STR_DESC[0];
2601
2602 3528 2 RSL_LEN[0] = .SCB[PSMSL_PAGE_WIDTH] - 20;      | set the page length
2603 3529 2 3530 2 3531 2 3532 2 3533 2 3534 2 3535 2 3536 2 3537 2 3538 2 3539 1 WHILE CH$NEQ( 1, .STR_PTR, 1, CH$PTR(UPLIT(' ')))      | largest less than 180
2604 3529 2 3530 2 3531 2 3532 2 3533 2 3534 2 3535 2 3536 2 3537 2 3538 2 3539 1 DO ! trim off chars until blanks
2605 3530 2 3531 2 3532 2 3533 2 3534 2 3535 2 3536 2 3537 2 3538 2 3539 1 BEGIN
2606 3531 2 3532 2 3533 2 3534 2 3535 2 3536 2 3537 2 3538 2 3539 1 RSL_LEN[0] = .RSL_LEN[0] - 1;
2607 3532 2 3533 2 3534 2 3535 2 3536 2 3537 2 3538 2 3539 1 STR_PTR = CH$PTR(.STR_DESC[ADDR]+.RSL_LEN[0]);
2608 3533 2 3534 2 3535 2 3536 2 3537 2 3538 2 3539 1 END;
2609 3534 2 3535 2 3536 2 3537 2 3538 2 3539 1 END;
2610
2611
2612
2613
2614

```

20	20	53	4D	56	2F	58	41	56	01136	P.AAR:	.ASCII	\VAX/VMS
20	20	53	4D	56	2F	58	41	56	0113F		.ASCII	\VAX/VMS
20	20	53	4D	56	2F	58	41	56	01148		.ASCII	\VAX/VMS
20	20	53	4D	56	2F	58	41	56	01151		.ASCII	\VAX/VMS
20	20	53	4D	56	2F	58	41	56	0115A		.ASCII	\VAX/VMS
20	20	53	4D	56	2F	58	41	56	01163		.ASCII	\VAX/VMS
20	20	53	4D	56	2F	58	41	56	0116C		.ASCII	\VAX/VMS
20	20	53	4D	56	2F	58	41	56	01175		.ASCII	\VAX/VMS
20	20	53	4D	56	2F	58	41	56	0117E		.ASCII	\VAX/VMS
20	20	53	4D	56	2F	58	41	56	01187		.ASCII	\VAX/VMS
20	20	53	4D	56	2F	58	41	56	01190		.ASCII	\VAX/VMS
20	20	53	4D	56	2F	58	41	56	01199		.ASCII	\VAX/VMS
20	20	53	4D	56	2F	58	41	56	011A2		.ASCII	\VAX/VMS
20	20	53	4D	56	2F	58	41	56	011AB		.ASCII	\VAX/VMS
20	20	53	4D	56	2F	58	41	56	011B4		.ASCII	\VAX/VMS
20	20	53	4D	56	2F	58	41	56	011BD		.ASCII	\VAX/VMS
20	20	53	4D	56	2F	58	41	56	011C6		.ASCII	\VAX/VMS
20	20	53	4D	56	2F	58	41	56	011CF		.ASCII	\VAX/VMS
20	20	53	4D	56	2F	58	41	56	011D8		.ASCII	\VAX/VMS
20	20	53	4D	56	2F	58	41	56	011E1		.ASCII	\VAX/VMS
20	20	53	4D	56	2F	58	41	56	011EA		.ASCII	\VAX/VMS
20	20	53	4D	56	2F	58	41	56	011F3		.ASCII	\VAX/VMS
20	20	53	4D	56	2F	58	41	56	011FC		.ASCII	\VAX/VMS
20	20	53	4D	56	2F	58	41	56	01205		.ASCII	\VAX/VMS
		53	4D	56	2F	58	41	56	0120E		.ASCII	\VAX/VMS\

SEPARATE
V04-001Print Symbiant -- separation routines
GET_VMS_LOGO - [Create a Phrase of VMS logo]K 3
16-Sep-1984 02:23:03
14-Sep-1984 22:32:26VAX-11 Bliss-32 V4.0-742
[PRTSMB.SRC]SEPARATE.B32;2Page 84
(23)

00	00	00	20	01215	.BLKB	3
				01218	P.AAQ:	.LONG 223
				00000000	0121C	.ADDRESS P.AAR
				01220	P.AAS:	.ASCII \ <0><0><0>

TRAILING= 1
DEFAULT= P.AAQ

001C 00000 GET_VMS_LOGO:						
						.WORD Save R2,R3,R4
53	14	04	AC D0 00002	MOVL SCB, R3		: 3476
		0200	C3 D1 00006	CMPL 512(R3), #20		: 3516
			04 18 0000B	BGEQ 1\$		
			0C BC B4 0000D	CLRW @RSL_LEN		
			04 00010	RET		
54	08	AC D0 00011	1\$:	MOVL STR_DESC, R4		
		54 DD 00015		PUSHL R4		
52	0C	AC D0 00017		MOVL RSL_LEN, R2		
		52 DD 0001B		PUSHL R2		
		D4 AF 9F 0001D		PUSHAB DEFAULT		
62	00000000G	00	03 FB 00020	CALLS #3, SYSSFAO		
	0200	C3	14 A3 00027	SUBW3 #20, 512(R3), (R2)		
		50	62 3C 0002D	MOVZWL (R2), STR_PTR		
		50	04 A4 C0 00030	ADDL2 4(R4), STR_PTR		
		C4 AF	60 91 00034	(CMPB (STR_PTR), -P.AAS		
			04 13 00038	BEQL 3\$		
			62 B7 0003A	DECW (R2)		
			EF 11 0003C	BRB 2\$		
			04 0003E	3\$:	RET	

: 3519
: 3518
: 3526
: 3528
: 3530
: 3532
: 3535
: 3536
: 3539

: Routine Size: 63 bytes, Routine Base: CODE + 1224

```

2616      3540 1 %sbttl 'GET_DIGITAL_LOGO - Create a Phrase of Digital logo'
2617      3541 1 ++
2618      3542 1 | Functional Description:
2619      3543 1 |   VAX/VMS Version Vx.x
2620      3544 1 |
2621      3545 1 | Formal Parameters:
2622      3546 1 |   SCB          - Address of the SCB
2623      3547 1 |   STR_DESC    - Desc of String to Return
2624      3548 1 |   RET_LEN     - Return length of Desc.
2625      3549 1 |
2626      3550 1 | Implicit Inputs:
2627      3551 1 |   none
2628      3552 1 |
2629      3553 1 | Implicit Outputs:
2630      3554 1 |   none
2631      3555 1 |
2632      3556 1 | Returned Value:
2633      3557 1 |   none
2634      3558 1 |
2635      3559 1 | Side Effects:
2636      3560 1 |   none
2637      3561 1 |
2638      3562 1 ROUTINE GET_DIGITAL_LOGO (
2639      3563 1 |   SCB          : REF $BBLOCK,           ! SCB
2640      3564 1 |   STR_DESC    : REF VECTOR[2],        ! Output buffer desc
2641      3565 1 |   RSL_LEN     : REF VECTOR [,WORD]   ! Return length (word)
2642      3566 1 |   ) : NOVALUE =
2643      3567 2 BEGIN
2644      3568 2 BIND
2645      P 3569 2 DEFAULT = $DESCRIPTOR (
2646      P 3570 2 |   '!AC - VAX/VMS Version ',
2647      P 3571 2 |   '!AS');
2648      3572 2 |
2649      3573 2 LOCAL
2650      3574 2 |   LOGO
2651      3575 2 |   FAO_DESC   : VECTOR[2],
2652      3576 2 |   BUFFER     : VECTOR[20,byte],
2653      3577 2 |   ITEM_LIST  : $ITMLST_DECL (ITEMS=1);
2654      3578 2 |
2655      3579 2 |   FAO_DESC[SIZE] = %ALLOCATION(BUFFER);
2656      3580 2 |   FAO_DESC[ADDR] = BUFFER;
2657      3581 2 |
2658      3582 2 |   IF .SCB[PSMSL_PAGE_WIDTH] LSS 52           ! 52 chars in complete logo
2659      3583 2 |   THEN
2660      3584 2 |   |   LOGO = UPLIT BYTE (%ASCIC 'DEC')
2661      3585 2 |   ELSE
2662      3586 2 |   |   LOGO = UPLIT BYTE (%ASCIC 'Digital Equipment Corporation');
2663      3587 2 |
2664      P 3588 2 |   $ITMLST_INIT (ITMLST=ITEM_LIST,
2665      P 3589 2 |   (
2666      P 3590 2 |   |   ITMCOD=SYIS_VERSION,
2667      P 3591 2 |   |   BUFADR=.FAO_DESC[ADDR],
2668      P 3592 2 |   |   BUFSIZ=8,
2669      P 3593 2 |   |   RETLEN=FAO_DESC[SIZE]
2670      P 3594 2 |   |   ));
2671      P 3595 2 |   $GETSYIW(ITMLST=ITEM_LIST);
2672      3596 2

```

```

: 2673 3597 2
: 2674 3598 2 FAO_DESC[SIZE] = DELIMIT_STRING_NOT (.FAO_DESC[ADDR],
: 2675 3599 2                                     XCHAR(32), .FAO_DESC[SIZE]);
: 2676 P 3600 2 $FAO (
: 2677 P 3601 2     DEFAULT,
: 2678 P 3602 2     RSL_LEN[0],
: 2679 P 3603 2     STR_DESC[0],
: 2680 P 3604 2     LOGO,
: 2681 P 3605 2     FAO_DESC[0])
: 2682 3606 3 ! version
: 2683 3607 1 END;

```

56 20 53 4D 56 2F 58 41 56 20 20 2D 20 69	43 41 21 01263 P.AAU:	.ASCII \!AC - VAX/VMS Version \	:
60 70 69 75 71 45 20 6C 61 74 69 67 69	73 72 65 01272 53 41 21 01279 00000019 0127C P.AAT:	.ASCII \!AS\	6F
6E 6F 69 74 61 72 6F 70 72 6F 43 20	00000000' 01280 43 45 44 03 01284 P.AAV:	.LONG 25	52
	69 44 1D 01288 P.AAU:	.ADDRESS P.AAU	6E
	74 6E 65 01297	.ASCII <3>\DEC\	6F
		.ASCII <29>\Digital Equipment Corporation\	

DEFAULT= P.AAT
.EXTRN SYSSGETSYIW

0004 00000 GET_DIGITAL_LOGO:					
24 5E	2C C2 00002	.WORD	Save R2	3562	
28 AE	14 D0 00005	SUBL2	#4, SP	3579	
28 AE	AE 9E 00009	MOVL	#20, FAO_DESC	3580	
50 04	AC D0 0000E	MOVAB	BUFFER FAO_DESC+4	3582	
34 0200	CO D1 00012	MOVL	SCB, R0		
	06 18 00017	CMPL	512(R0), #52		
52 C2	AF 9E 00019	BGEQ	1S	3584	
	04 11 0001D	MOVAB	P.AAV, LOGO		
52 C0	AF 9E 0001F	BRB	2\$		
50 10000008	6E 9E 00023	1\$: MOVAB	P.AAW, LOGO	3586	
80 28	8F D0 00026	ITEM LIST	\$SITMBLKPTR	3594	
80 24	AE D0 0002D	MOVL	#268435464, (\$SITMBLKPTR)+		
80 24	AE 9E 00031	FAO_DESC+4, (\$SITMBLKPTR)+			
	80 D4 00035	MOVAB	FAO_DESC, (\$SITMBLKPTR)+		
	7E 7C 00037	CLRL	(\$SITMBLKPTR)+		
	7E D4 00039	CLRQ	-(SP)	3596	
0C	AE 9F 0003B	CLRL	-(SP)		
00000000G 00	7E 7C 0003E	PUSHAB	ITEM LIST		
	7E D4 00040	CLRL	-(SP)		
	07 FB 00042	CALLS	#7, SYSSGETSYIW		
24	AE DD 00049	PUSHL	FAO_DESC	3599	
	20 DD 0004C	PUSHL	#32	3598	
0000V CF	30 AE DD 0004E	PUSHL	FAO_DESC+4		
24 AE	03 FB 00051	CALLS	#3, DELIMIT_STRING_NOT		
	50 D0 00056	MOVL	R0, FAO_DESC		
	24 AE 9F 0005A	PUSHAB	FAO_DESC		
	52 DD 0005D	PUSHL	LOGO	3605	
	08 AC DD 0005F	PUSHL	STR_DESC		
	0C AC DD 00062	PUSHL	RSL_LEN		

SEPARATE
V04-001

Print Symbiont -- separation routines
GET_DIGITAL_LOGO - Create a Phrase of Digital N 3
16-Sep-1984 02:23:03 VAX-11 Bliss-32 V4.0-742
[PRTSMB.SRC]SEPARATE.B32;2

Page 87
(24)

SEI
VO4

00000000G 00 FF6D CF 9F 00065
05 FB 00069
04 00070 PUSHAB DEFAULT
CALLS #5, SYSSFAO
RET

::
:: 3607

: Routine Size: 113 bytes. Routine Base: CODE + 12A6

```

: 2685      3608 1 %sbttl 'GET_JOB_DESCRIPTION - Create a Sentence Describing the Current Job'
: 2686      3609 1 ++
: 2687      3610 1 | Functional Description:
: 2688      3611 1 | This routine creates a sentence describing the current job.
: 2689      3612 1 |
: 2690      3613 1 | Formal Parameters:
: 2691      3614 1 |   SCB          = Address of the SCB
: 2692      3615 1 |   STR_DESC     = Desc of String to Return
: 2693      3616 1 |   RET_LEN      = Return Length of Desc.
: 2694      3617 1 |
: 2695      3618 1 | Implicit Inputs:
: 2696      3619 1 |   none
: 2697      3620 1 |
: 2698      3621 1 | Implicit Outputs:
: 2699      3622 1 |   none
: 2700      3623 1 |
: 2701      3624 1 | Returned Value:
: 2702      3625 1 |   none
: 2703      3626 1 |
: 2704      3627 1 | Side Effects:
: 2705      3628 1 |   none
: 2706      3629 1 | --
: 2707      3630 1 ROUTINE GET_JOB_DESCRIPTION (
: 2708      3631 1 |   SCB          : REF $BBLOCK,           ! SCB
: 2709      3632 1 |   TIME_FLAG    : REF VECTOR[2],       ! Output buffer desc
: 2710      3633 1 |   STR_DESC     : REF VECTOR [,WORD]   ! Return length (word)
: 2711      3634 1 |   RET_LEN      : NOVALUE =
: 2712      3635 1 |
: 2713      3636 2 BEGIN
: 2714      3637 2 BIND
: 2715      3638 2 TRAILING = 1,
: 2716      3639 2 LEADING = 0,
: 2717      3640 2 |
: 2718      P 3641 2 NODE = $DESCRIPTOR(
: 2719      P 3642 2 |   'SYSSNODE'),! - system annoucement
: 2720      P 3643 2 |
: 2721      P 3644 2 DATE_FORMAT = $DESCRIPTOR(
: 2722      P 3645 2 |   '!17%D'),
: 2723      P 3646 2 |
: 2724      P 3647 2 SENT_FORMAT1 = $DESCRIPTOR(
: 2725      P 3648 2 |   'Job !AS ',
: 2726      P 3649 2 |   '(!UL',
: 2727      P 3650 2 |   'queued to !AS ',
: 2728      P 3651 2 |   'on !AS',
: 2729      P 3652 2 |   'by user !AS, ',
: 2730      P 3653 2 |   'UIC !%I',
: 2731      P 3654 2 |   'under account !AS',
: 2732      P 3655 2 |   'at priority !UL',
: 2733      P 3656 2 |   '!AC',
: 2734      P 3657 2 |
: 2735      P 3658 2 |   'on printer !AS',
: 2736      P 3659 2 |   'on !AS',
: 2737      P 3660 2 |   'from queue !AS',
: 2738      P 3661 2 |   '..');
: 2739      P 3662 2 |
: 2740      P 3663 2 LOCAL
: 2741      P 3664 2 |   RET_LENGTH .

```

- job name
 - job number
 - batch file name(pres tense)
 - time queued
 - user name
 - user uic
 - user account
 - que priority
 - 'started'/'completed'/
 - 'restarted'/'aborted'
 - device name
 - time printed
 - executor queue
 - period

```
: 2742      3665 2 STATUS
: 2743      3666 2 DOUBLE_COLONS
: 2744      3667 2 CHOICE
: 2745      3668 2 DATE_QUEUED : VECTOR[2]
: 2746      3669 2 DATE_QUEUED_BUFF : VECTOR[17,byte]
: 2747      3670 2 DATE_PRINTED : VECTOR[2]
: 2748      3671 2 DATE_PRINTED_BUFF : VECTOR[17,byte]
: 2749      3672 2 ACCOUNT_DESC : VECTOR[2]
: 2750      3673 2 USERNAME_DESC : VECTOR[2]; ! desc of string
: 2751
: 2752      3674 2 ! get the user name delimited
: 2753      3675 2
: 2754      3677 2 USERNAME_DESC[SIZE] = .SCB_SIZE_(USER_NAME);
: 2755      3678 2 USERNAME_DESC[ADDR] = .SCB_ADDR_(USER_NAME);
: 2756      3679 2 ! Insert only the string ... No trailing blanks
: 2757      3680 2
: 2758      3681 2 DISCARD (TRAILING, XC' ', USERNAME_DESC[ADDR], .USERNAME_DESC[SIZE],
: 2759      3682 2           USERNAME_DESC[SIZE],USERNAME_DESC[ADDR]); !Return length and pointer
: 2760
: 2761      3683 2
: 2762      3684 2 ! get the account name delimited
: 2763      3685 2
: 2764      3686 2 ACCOUNT_DESC[SIZE] = .SCB_SIZE_(ACCOUNT_NAME);
: 2765      3687 2 ACCOUNT_DESC[ADDR] = .SCB_ADDR_(ACCOUNT_NAME);
: 2766      3688 2 ! Insert only the string ... No trailing blanks
: 2767      3689 2
: 2768      3690 2 DISCARD (TRAILING, XC' ', ACCOUNT_DESC[ADDR], .ACCOUNT_DESC[SIZE],
: 2769      3691 2           ACCOUNT_DESC[SIZE],ACCOUNT_DESC[ADDR]); ! Return length and pointer
: 2770      3692 2
: 2771      3693 2 ! start, restart and complete
: 2772      3694 2 IF .TIME_FLAG
: 2773      3695 2 THEN
: 2774      3696 2 BEGIN
: 2775      3697 2   CHOICE = UPLIT BYTE (%ASCII 'started');
: 2776      3698 2   IF .REQUEST_FLAG_(RESTARTING)
: 2777      3699 2   THEN
: 2778      3700 2     CHOICE = UPLIT BYTE (%ASCII 'restarted');
: 2779      3701 2 END
: 2780      3702 2 ELSE
: 2781      3703 2 BEGIN
: 2782      3704 2   BIND CONDITION = SCB[PSMST_CONDITION_AREA] : VECTOR; ! Task completion status
: 2783      3705 2
: 2784      3706 2   ! Assume job completed normally
: 2785      3707 2
: 2786      3708 2   CHOICE = UPLIT BYTE (%ASCII 'completed');
: 2787      3709 2
: 2788      3710 2   ! Check completion status for an error
: 2789      3711 2
: 2790      3712 2   IF .CONDITION[0] NEQU 0
: 2791      3713 2   THEN
: 2792      3714 2   BEGIN
: 2793      3715 2     ! Assume job controller or symbiont initiated abort
: 2794      3716 2
: 2795      3717 2     CHOICE = UPLIT BYTE (%ASCII 'ABORTED');
: 2796      3718 2
: 2797      3719 2     ! Check for special case of job controller initiated requeue
: 2798      3720 2
: 2799      3721 2
```

D 4
16-Sep-1984 02:23:03 VAX-11 Bliss-32 V4.0-742
GET_JOB_DESCRIPTION - Create a Sentence Describ 14-Sep-1984 22:32:26 [PRTSMB.SRC]SEPARATE.B32;2

```

2799 3722 4 !*| FEATURE DISABLED UNTIL JOB CONTROLLER MESSAGE AVAILABILITY STRAIGHTENED OUT
2800 3723 4 !*| IF .CONDITION[0] EQLU JBCS_JOBQUEUE
2801 3724 4 !*| THEN
2802 3725 4 !*|     CHOICE = UPLIT BYTE (%ASCIC 'REQUEUED');
2803 3726
2804 3727 3     END;
2805 3728 2     END;
2806 3729
2807 3730 2 ! Get and delimit the date/times
2808 3731 2 ! time queued
2809 3732 2 DATE_QUEUED[SIZE] = %ALLOCATION(DATE_QUEUED_BUFF);
2810 3733 2 DATE_QUEUED[ADDR] = DATE_QUEUED_BUFF;
2811 3734
2812 P 3735 2 $FAO ( DATE_FORMAT,
2813 P 3736 2        RET [LENGTH,
2814 P 3737 2        DATE_QUEUED[0],
2815 P 3738 2        SCB[PSMSQ_TIME_QUEUED]);
2816 3739 2 !
2817 3740 2 DISCARD (LEADING, %C' ', DATE_QUEUED[ADDR], .RET_LENGTH,
2818 3741 2            DATE_QUEUED[SIZE], DATE_QUEUED[ADDR]); ! Return length and pointer
2819 3742 2 DISCARD (TRAILING, %C' ', DATE_QUEUED[ADDR], .RET_LENGTH,
2820 3743 2            DATE_QUEUED[SIZE], DATE_QUEUED[ADDR]); ! Return length and pointer
2821 3744
2822 3745 2 ! time printed
2823 3746 2 DATE_PRINTED[SIZE] = %ALLOCATION(DATE_PRINTED_BUFF);
2824 3747 2 DATE_PRINTER[ADDR] = DATE_PRINTED_BUFF;
2825 3748
2826 P 3749 2 $FAO ( DATE_FORMAT,
2827 P 3750 2        RET [LENGTH,
2828 P 3751 2        DATE_PRINTED[0],
2829 P 3752 2        SCB[PSMSQ_TIME_PRINTED]);
2830 3753 2 !
2831 3754 2 DISCARD (LEADING, %C' ', DATE_PRINTED[ADDR], .RET_LENGTH,
2832 3755 2            DATE_PRINTED[SIZE], DATE_PRINTED[ADDR]); ! Return length and pointer
2833 3756 2 DISCARD (TRAILING, %C' ', DATE_PRINTED[ADDR], .RET_LENGTH,
2834 3757 2            DATE_PRINTED[SIZE], DATE_PRINTED[ADDR]); ! Return length and pointer
2835 3758
2836 P 3759 2 $FAO ( SENT_FORMAT1,
2837 P 3760 2        RET [LEN[0]],
2838 P 3761 2        STR_DESC[0];
2839 P 3762 2        SCB[PSMSQ_JOB_NAME],
2840 P 3763 2        .SCB[PSMSL_ENTRY_NUMBER],      | job name
2841 P 3764 2        SCB[PSMSQ_QUEUE],
2842 P 3765 2        DATE_QUEUED[0],                  | entry number
2843 P 3766 2        USERNAME_DESC[0],                | batch que present tense
2844 P 3767 2        .SCB[PSMSL_UIC],
2845 P 3768 2        ACCOUNT_DESC[0],                  | time queued
2846 P 3769 2        .SCB[PSMSL_PRIORITY],            | user name
2847 P 3770 2        .CHOICE,                          | user uic
2848 P 3771 2        SCB[PSMSQ_DEVICE_NAME],
2849 P 3772 2        DATE_PRINTED[0],                  | user account
2850 P 3773 2        SCB[PSMSQ_EXECUTOR_QUEUE]      | queue priority
2851 3774 2        );
2852 3775 2        ;
2853 3776 2 RETURN SSS_NORMAL;
2854 3777 1 END;

```

SCB[PSMSQ_JOB_NAME],	job name
.SCB[PSMSL_ENTRY_NUMBER],	entry number
SCB[PSMSQ_QUEUE],	batch que present tense
DATE_QUEUED[0],	time queued
USERNAME_DESC[0],	user name
.SCB[PSMSL_UIC],	user uic
ACCOUNT_DESC[0],	user account
.SCB[PSMSL_PRIORITY],	queue priority
.CHOICE,	started/completed/restarted
SCB[PSMSQ_DEVICE_NAME],	device name
DATE_PRINTED[0],	time printed
SCB[PSMSQ_EXECUTOR_QUEUE]	executor queue

SEPARATE
V04-001Print Symbiont -- separation routines
GET_JOB_DESCRIPTION - Create a Sentence DescribE 4
16-Sep-1984 02:23:03
14-Sep-1984 22:32:26VAX-11 Bliss-32 V4.0-742
[PRTSMB.SRC]SEPARATE.B32;2Page 91
(25)

45	44	4F	4E	24	53	59	53	01317	P.AAY:	.ASCII \SYSSNODE\	
					00000008,	0131F		.BLKB	1		
					00000000,	01320	P.AAX:	.LONG	8		
				44	25	37	31	21	01324	.ADDRESS P.AAY	
								01328	P.ABA:	.ASCII \!17%D\	
						00000005,	0132D	.BLKB	3		
						00000000,	01330	P.AAZ:	.LONG	5	
							01334	.ADDRESS P.ABA			
				20	53	41	21	20	01338	.ASCII \Job !AS \	
20	53	41	21	20	6F	74	20	64	.ASCII \(!UL) \		
					20	53	41	29	.ASCII \queued to !AS \		
					20	53	41	21	.ASCII \on !AS \		
					20	72	65	73	.ASCII \by user !AS, \		
21	20	74	6E	75	6F	63	63	61	20	.ASCII \UIC !%, \	
					20	2C	49	25	21	.ASCII \under account !AS \	
					20	72	65	64	20	.ASCII \at priority !UL. \	
4C	55	21	20	79	74	69	72	6F	69	01380	
					72	70	70	20	74	.ASCII \!AC \	
20	53	41	21	20	72	65	74	6E	69	.ASCII \on printer !AS \	
					20	53	41	21	20	.ASCII \on !AS \	
53	41	21	20	65	75	65	75	20	6E	.ASCII \from queue !AS\	
					75	71	20	6D	6F	.ASCII \.\	
							013BC	.BLKB	3		
							013BD	.LONG	133		
						00000085,	013C0	P.ABB:	.ADDRESS P.ABC		
						00000000,	013C4	P.ABD:	.ASCII <7>\started\		
64	65	64	65	74	72	61	74	73	07	013C8	.ASCII <9>\restarted\
64	65	74	65	6C	72	61	74	72	09	013D0	.ASCII <9>\completed\
		44	45	54	52	4F	42	41	07	013DA	.ASCII <7>\ABORTED\
							013E4	P.ABG:	.ASCII		

TRAILING= 1
LEADING= 0
NODE= P.AAX
DATE_FORMAT= P.AAZ
SENT_FORMAT1= P.ABB

003C 00000 GET_JOB_DESCRIPTION:

55	0000V	CF	9E	00002	.WORD	Save R2,R3,R4,R5	3630
54	0000000G	00	9E	00007	MOVAB	DISCARD, R5	
5E	B4	AE	9E	0000E	MOVAB	SYSSFAO, R4	
52	04	AC	D0	00012	MOVAB	-76(SP), SP	3677
50	016C	C2	9E	00016	MOVL	SCB, R2	
04	AE	60	3C	0001B	MOVAB	364(R2), R0	
08	AE	04	A0	0001F	MOVL	(R0), USERNAME DESC	
		08	AE	9F 00024	PUSHAB	4(R0), USERNAME_DESC+4	3678
		08	AE	9F 00027	PUSHAB	USERNAME DESC	3682
		0C	AE	DD 0002A	PUSHL	USERNAME DESC	
		14	AE	DD 0002D	PUSHL	USERNAME DESC	3681
			20	DD 00030	PUSHL	USERNAME DESC+4	
			01	DD 00032	PUSHL	#32	
			06	FB 00034	PUSHL	#1	
65	50	1C	A2	9E 00037	CALLS	#6, DISCARD	
					MOVAB	28(R2), R0	3686

SEPARATE
V04-001

Print Symbiont -- separation routines
GET_JOB_DESCRIPTION - Create a Sentence

F 4
16-Sep-1984 02:23:03
14-Sep-1984 22:32:26

VAX-11 Bliss-32 V4.0-742
[PRTSMB.SRC]SEPARATE.B32;2

Page 92
(25)

SEPARATE
V04-001

Print Symbiont -- separation routines
GET_JOB_DESCRIPTION - Create a Sentence Describ

G 4
16-Sep-1984 02:23:03
14-Sep-1984 22:32:26

VAX-11 Bliss-32 V4.0-742
[PRTSMB.SRC]SEPARATE.B32;2

Page 93
(25)

SE
VO

65	20 DD 000F6	PUSHL #32
	01 DD 000F8	PUSHL #1
	06 FB 000FA	CALLS #6, DISCARD
	5C A2 9F 000FD	PUSHAB 92(R2)
	2C AE 9F 00100	PUSHAB DATE PRINTED
	4C A2 9F 00103	PUSHAB 76(R2)
	53 DD 00106	PUSHL CHOICE
	0128 C2 DD 00108	PUSHL 296(R2)
	20 AE 9F 0010C	PUSHAB ACCOUNT_DESC
	0168 C2 DD 0010F	PUSHL 360(R2)
	20 AE 9F 00113	PUSHAB USERNAME_DESC
	64 AE 9F 00116	PUSHAB DATE queued
	012C C2 9F 00119	PUSHAB 300(R2)
	58 A2 DD 0011D	PUSHL 88(R2)
	00A8 C2 9F 00120	PUSHAB 168(R2)
	0C AC DD 00124	PUSHL STR_DESC
	10 AC DD 00127	PUSHL RET_LEN
64	FEA6 CF 9F 0012A	PUSHAB SENT_FORMAT1
	0F FB 0012E	CALLS #15, SYSSFAO
	04 00131	RET

3774

3777

; Routine Size: 306 bytes. Routine Base: CODE + 13EC

SEPARATE
V04-001

Print Symbiont -- separation routines 16-Sep-1984 02:23:03 VAX-11 Bliss-32 v4.0-742
GET FILE DESCRIPTION - Create a Sentence Descri 16-Sep-1984 22:32:26 [PRISMB.SRC]SEPARATE.B32.2

H 4

16-Sep-1984 02:23:03

VAX-11 Bliss-32 V4.0-742
[PRTSMB.SRC]SEPARATE.B32:2

Page 94
(26)

SE
VO

```

2856 3778 1 %sbttl 'GET_FILE_DESCRIPTION - Create a Sentence Describing the Current File'
2857 3779 1 ++
2858 3780 1 Functional Description:
2859 3781 1 This routine creates a sentence describing the current file.
2860 3782 1
2861 3783 1 Formal Parameters:
2862 3784 1 SCB           - Address of the SCB
2863 3785 1 STR_DESC     - Desc of String to Return
2864 3786 1 RET_LEN      - Return Length of Desc.
2865 3787 1
2866 3788 1 Implicit Inputs:
2867 3789 1             none
2868 3790 1
2869 3791 1 Implicit Outputs:
2870 3792 1             none
2871 3793 1
2872 3794 1 Returned Value:
2873 3795 1             none
2874 3796 1
2875 3797 1 Side Effects:
2876 3798 1             none
2877 3799 1 --
2878 3800 1 ROUTINE GET_FILE DESCRIPTION  (
2879 3801 1     SCB          : REF $BLOCK,          ! SCB
2880 3802 1     STR_DESC    : REF VECTOR[2],       ! Output buffer desc
2881 3803 1     RET_LEN     : REF VECTOR [,WORD]   ! Return length (word)
2882 3804 1             ) : NOVALUE =
2883 3805 2 BEGIN
2884 3806 2 BIND
2885 3807 2     FAB = .SCB[PSMSA_FAB]: $BLOCK,
2886 3808 2     NAM = .SCB[PSMSA_NAM]: $BLOCK,
2887 3809 2     XABDAT = .SCB[PSMSA_XABDAT]: $BLOCK,
2888 3810 2     XABFHC = .SCB[PSMSA_XABFHC]: $BLOCK,
2889 3811 2     XABPRO = .SCB[PSMSA_XABPRO]: $BLOCK,
2890 3812 2
2891 P 3813 2 FORMAT POS = $DESCRIPTOR (
2892 P 3814 2     'File !AS ',                      ! - file name
2893 P 3815 2     '(!UL,!UL,!UL), ',                 ! - file Id number
2894 P 3816 2     'Last revised on !17%D, ',        ! - revision date
2895 P 3817 2     'is a !UL block ',                  ! - file size
2896 P 3818 2     '!AC file ',                     ! - file organization
2897 P 3819 2     'owned by UIC !%I. ',            ! - owner user uic
2898 P 3820 2
2899 P 3821 2 RECORD FORMAT = $DESCRIPTOR (
2900 P 3822 2     'The records are ',                   ! - record format
2901 P 3823 2     '!AC with '),
2902 P 3824 2
2903 P 3825 2 RECORD_VFC_FORMAT = $DESCRIPTOR (
2904 P 3826 2     'The records are ',                   ! - variable length
2905 P 3827 2     'variable length with a ',           ! - fixed control area size
2906 P 3828 2     'fixed control size of !UL byte!%S and '),
2907 P 3829 2
2908 P 3830 2 REC_SIZE = $DESCRIPTOR (
2909 P 3831 2     'The longest record is !UL byte!%S.', ! - max record size
2910 P 3832 2
2911 P 3833 2
2912 P 3834 2

```

SEPARATE
V04-001

Print Symbiont -- separation routines
GET_FILE_DESCRIPTION - [Create a Sentence Descri

16-Sep-1984 02:23:03
14-Sep-1984 22:32:26

VAX-11 Bliss-32 V4.0-742
[PRTSMB.SRC]SEPARATE.B32;2

Page 95
(26)

2913 P 3835 2 CARRIAGE_FORMAT = \$DESCRIPTOR (!AC').
2914 3836 2 ! - record attributes
2915 3837 2
2916 P 3838 2 FORMAT NEG = \$DESCRIPTOR ('File (!AS) description is unavailable to the symbiont.');
2917 3839 2
2918 3840 2
2919 3841 2 LITERAL
2920 3842 2 K_MAX_BUFFER_SIZE = 512;
2921 3843 2
2922 LOCAL
2923 3844 2 RECORD_SIZE .
2924 3845 2 FILE_SIZE .
2925 3846 2 ORGANIZATION .
2926 3847 2 ATTRIBUTES .
2927 3848 2 FORMAT .
2928 3849 2 CURRENT_LEN : INITIAL (0).
2929 3850 2 DATE_REVISED : VECTOR[2].
2930 3851 2 STRING_PTR : VECTOR [2]; ! Pointer to current string
2931 3852 2
2932 3853 2 ! Allocate the buffer for "GET_xxx" Routines
2933 3854 2
2934 3855 2 ! STRING_PTR[SIZE] = K_MAX_BUFFER_SIZE;
2935 3856 2 STRING_PTR[ADDR] = .STR_DESC[ADDR]; ! init address
2936 3857 2
2937 3858 2
2938 3859 2 RET_LEN[0] = 0;
2939 3860 2
2940 3861 2 IF FILE_OPEN(.SCB)
2941 3862 2 THEN
2942 3863 2 BEGIN
2943 3864 2 ! get the file size
2944 3865 2
2945 3866 3 FILE_SIZE = .XABFH[.XABSL_EBK];
2946 3867 3 IF (.XABFH[.XABSW_FFB] EQ[0] AND
2947 3868 4 (.FILE_SIZE NEQ 0))
2948 3869 4 THEN
2949 3870 4 FILE_SIZE = .FILE_SIZE - 1;
2950 3871 2
2951 3872 2 ! insert file organization
2952 3873 2
2953 3874 2 IF .FAB[FAB\$B_ORG] EQL FAB\$C_IDX
2954 3875 2 THEN
2955 3876 2 ! ORGANIZATION = UPLIT BYTE (%ASCIC 'indexed')
2956 3877 2 ELSE IF .FAB[FAB\$B_ORG] EQL FAB\$C_SEQ
2957 3878 2 THEN
2958 3879 2 ! ORGANIZATION = UPLIT BYTE (%ASCIC 'sequential')
2959 3880 2 ELSE IF .FAB[FAB\$B_ORG] EQL FAB\$C_REL
2960 3881 2 THEN
2961 3882 2 ! ORGANIZATION = UPLIT BYTE (%ASCIC 'relative')
2962 3883 2 ELSE
2963 3884 2 ! ORGANIZATION = UPLIT BYTE (%ASCIC 'undefined organization');
2964 3885 2
2965 3886 2
2966 P 3887 3 \$FAO (FORMAT_POS,
2967 P 3888 3 CURRENT_LEN,
2968 P 3889 3 STRING_PTR[0],
2969 P 3890 3 SCB[PSMSQ FILE_SPECIFICATION],
P 3891 3 .NAM[NAMSD_FID_NUM],

2970 P 3892 J
2971 P 3893 .NAM[NAM\$W_FID_SEQ],
2972 P 3894 .NAM[NAM\$W_FID_RVN],
2973 P 3895 XABDAT[XAB\$Q_RDT].
2974 P 3896 .FILE_SIZE,
2975 P 3897 .ORGANIZATION,
2976 P 3898 XABPRO[XABSL_UIC]
2977 P 3899):
2978 P 3900 RET_LEN[0] = .RET_LEN[0] + .CURRENT_LEN;
2979 P 3901 STRING_PTR[ADDR] = .STRING_PTR[ADDR] + .CURRENT_LEN;
2980 P 3902 STRING_PTR[SIZE] = K_MAX_BUFFER_SIZE - .RET_LEN[0];
2981 P 3903 ! get record format
2982 P 3904
2983 P 3905
2984 P 3906
2985 P 3907 RECORD_SIZE = .XABFHG[XABSW_LRL]; ! record size
2986 P 3908
2987 P 3909 IF .FAB[FAB\$B_RF] NEQ FAB\$C_VFC
2988 P 3910 THEN
2989 P 3911 BEGIN
2990 P 3912 ! get record type
2991 P 3913
2992 P 3914 SELECTONE .FAB[FAB\$B_RF] OF
2993 P 3915 SET
2994 P 3916 [FAB\$C_FIX]: FORMAT = UPLIT BYTE
2995 P 3917 (%ASCIC 'fixed-length');
2996 P 3918 [FAB\$C_STM]: FORMAT = UPLIT BYTE
2997 P 3919 (%ASCIC 'stream');
2998 P 3920 [FAB\$C_STMCR]: FORMAT = UPLIT BYTE
2999 P 3921 (%ASCIC 'stream-CR');
3000 P 3922 [FAB\$C_STMLF]: FORMAT = UPLIT BYTE
3001 P 3923 (%ASCIC 'stream-LF');
3002 P 3924 [FAB\$C_UDF]: FORMAT = UPLIT BYTE
3003 P 3925 (%ASCIC 'an undefined format');
3004 P 3926 [FAB\$C_VAR]: FORMAT = UPLIT BYTE
3005 P 3927 (%ASCIC 'variable length');
3006 P 3928 TES:
3007 P 3929
3008 P 3930 \$FAO (RECORD_FORMAT,
3009 P 3931 CURRENT_LEN,
3010 P 3932 STRING_PTR[0],
3011 P 3933 .FORMAT
3012 P 3934):
3013 P 3935
3014 P 3936 END
3015 P 3937 ELSE
3016 P 3938 \$FAO (RECORD_VFC_FORMAT,
3017 P 3939 CURRENT_LEN,
3018 P 3940 STRING_PTR[0],
3019 P 3941 .FAB[FAB\$B_FSZ]
3020 P 3942):
3021 P 3943
3022 P 3944 RET_LEN[0] = .RET_LEN[0] + .CURRENT_LEN;
3023 P 3945 STRING_PTR[ADDR] = .STRING_PTR[ADDR] + .CURRENT_LEN;
3024 P 3946 STRING_PTR[SIZE] = K_MAX_BUFFER_SIZE - .RET_LEN[0];
3025 P 3947 ! get carriage control
3026 P 3948

3027 3949 3 !
3028 3950 3 IF .FAB[FAB\$V_(R)]
3029 3951 3 THEN ATTRIBUTES = UPLIT BYTE (%ASCIC 'implied (CR) carriage control')
3030 3952 3 ELSE BEGIN
3031 3953 3 IF .FAB[FAB\$V_FTN]
3032 3954 4 THEN ATTRIBUTES = UPLIT BYTE (%ASCIC 'FORTRAN (FTN) carriage control')
3033 3955 4 ELSE BEGIN
3034 3956 4 IF .FAB[FAB\$V_PRN]
3035 3957 4 THEN ATTRIBUTES = UPLIT BYTE (%ASCIC 'print file (PRN) carriage control')
3036 3958 4 ELSE ATTRIBUTES = UPLIT BYTE (%ASCIC 'imbedded (<none>) carriage control');
3037 3959 5 BEGIN
3038 3960 5 IF .FAB[FAB\$V_PRN]
3039 3961 5 THEN ATTRIBUTES =
3040 3962 5 UPLIT BYTE (%ASCIC 'print file (PRN) carriage control')
3041 3963 5 ELSE ATTRIBUTES = UPLIT BYTE (%ASCIC 'imbedded (<none>) carriage control');
3042 3964 5 END:
3043 3965 5 END:
3044 3966 5
3045 3967 4
3046 3968 3
3047 3969 3
3048 P 3970 3 SFAD {
3049 P 3971 3 CARRIAGE_FORMAT,
3050 P 3972 3 CURRENTLEN,
3051 P 3973 3 STRING_PTR[0],
3052 P 3974 3 .ATTRIBUTES);
3053 3975 3
3054 3976 3 RET_LEN[0] = .RET_LEN[0] + .CURRENT_LEN;
3055 3977 3 STRING_PTR[ADDR] = .STRING_PTR[ADDR] + .CURRENT_LEN;
3056 3978 3 STRING_PTR[SIZE] = K_MAX_BUFFER_SIZE - .RET_LEN[0];
3057 3979 3
3058 3980 3 IF .FAB[FAB\$B_RF] NEQ FABSC_FIX
3059 3981 3 THEN BEGIN
3060 3982 4 SFAD {
3061 P 3983 4 REC_SIZE,
3062 P 3984 4 CURRENTLEN,
3063 P 3985 4 STRING_PTR[0],
3064 P 3986 4 .RECORD_SIZE);
3065 3987 4
3066 3988 4
3067 3989 4 RET_LEN[0] = .RET_LEN[0] + .CURRENT_LEN;
3068 3990 4 STRING_PTR[ADDR] = .STRING_PTR[ADDR] + .CURRENT_LEN;
3069 3991 4 STRING_PTR[SIZE] = K_MAX_BUFFER_SIZE - .RET_LEN[0];
3070 3992 4 END;
3071 3993 3
3072 3994 3 END
3073 3995 2 ELSE BEGIN
3074 P 3996 3 SFAD {
3075 P 3997 3 FORMAT NEG,
3076 P 3998 3 CURRENTLEN,
3077 P 3999 3 STRING_PTR[0],
3078 P 4000 3 SCB[PSMSO_FILE_SPECIFICATION]
3079 P 4001 3);
3080 4002 3
3081 4003 3
3082 4004 2 RET_LEN[0] = .CURRENT_LEN;
3083 4005 2 END;

```

: 3084
: 3085
: 3086
: 3087
: 3088
: 3089
: 3090
: 3091
: 3092
: 3093
: 3094
4006 2
4007 2 ! final check for overflow
4008 2 IF .RET_LEN[0] GTR K_MAX_BUFFER_SIZE
4009 2 THEN
4010 3 BEGIN
4011 3 RET LEN[0] = 512;
4012 3 RETURN;
4013 2 END;
4014 2
4015 2 RETURN SSS_NORMAL;
4016 1 END;

```

20 2C 29 4C 55 21 20 53 41 21 20 65 6C 69 46 0151E P.ABI:	.ASCII \file !AS \
6E 6F 20 64 65 73 69 76 65 72 20 74 73 61 28 01527	.ASCII \ (!UL,!UL,!UL), \
20 6B 63 6F 6C 62 20 4C 55 21 20 61 20 73 69 01536	.ASCII \last revised on !17%D, \
25 21 20 43 49 55 20 79 62 20 64 65 6E 77 6F 01545	.ASCII \is a !UL block \
20 20 2E 49 0154D	.ASCII \!AC file \
25 21 20 43 49 55 20 79 62 20 64 65 6E 77 6F 0155C	.ASCII \owned by UIC !%I. \
20 20 2E 49 01565	
0000005A 01574	
00000000 01578 P.ABH:	.LONG 90
00000000 0157C P.ABK:	.ADDRESS P.ABI
65 72 61 20 73 64 72 6F 63 65 72 20 65 68 54 01580 P.ABK:	.ASCII \The records are \
20 68 74 69 77 20 43 41 21 0158F	
01590	.ASCII \!AC with \
01599	.BLKB 3
00000019 0159C P.ABJ:	.LONG 25
00000000 015A0 P.ABM:	.ADDRESS P.ABK
65 72 61 20 73 64 72 6F 63 65 72 20 65 68 54 015A4 P.ABM:	.ASCII \The records are \
68 74 67 6E 65 6C 20 65 6C 62 61 69 72 61 76 015B3	
20 61 20 68 74 69 77 20 43 41 21 015B4	.ASCII \variable length with a \
73 20 6C 6F 72 74 6E 6F 63 20 64 65 78 69 66 015C3	
65 74 79 62 20 4C 55 21 20 66 6F 20 65 7A 69 015CB	.ASCII \fixed control size of !UL byte!%S and \
20 64 6E 61 20 53 25 20 53 25 21 015DA	
015E9	
0000004D 015F1 P.ABL:	.BLKB 3
00000000 015F4 P.ABL:	.LONG 77
00000000 015F8 P.ABO:	.ADDRESS P.ABM
72 20 74 73 65 67 6E 6F 6C 69 20 65 68 54 20 20 015FC	.ASCII \ The longest record is !UL byte!%S.\
20 64 72 63 65 21 65 72 6F 63 65 0160B	
20 53 25 21 65 74 0161A	
00000024 01620 P.ABN:	.LONG 36
00000000 01624 P.ABN:	.ADDRESS P.ABO
2E 43 41 21 01628 P.ABQ:	.ASCII \!AC.\
00000004 0162C P.ABP:	.LONG 4
00000000 01630 P.ABQ:	.ADDRESS P.ABQ
63 73 65 64 20 29 53 41 21 28 20 65 6C 69 46 01634 P.ABS:	.ASCII \file (!AS) description is unavailable to\
76 61 6E 75 20 73 69 20 6E 6F 69 74 70 69 72 01643	
2E 74 6E 6F 69 62 60 79 73 20 65 68 74 20 01652	
0165C	.ASCII \ the symbiont.\
0166A	.BLKB 2
00000036 0166C P.ABR:	.LONG 54
00000000 01670 P.ABR:	.ADDRESS P.ABS
6C 61 69 64 65 78 65 64 6E 69 07 01674 P.ABT:	.ASCII <?>\indexed\
74 65 75 71 65 73 0A 0167C P.ABU:	.ASCII <10>\sequential\

SEPARATE
V04-001

Print_Symbiont -- separation routines
GET_FILE_DESCRIPTION - Create a Sentence

三

6-Sep-1984 02:23:03
4-Sep-1984 22:32:26

VAX-11 Bliss-32 V4.0-742
[PRTSMB.SRC]SEPARATE.B32:2

Page 99
(26)

61	67	72	6F	20	64	65	76	69	74	61	6C	65	72	08	01687	P.ABV:	.ASCII!	<8>\relative\
							6E	69	66	65	64	65	75	16	01690	P.ABW:	.ASCII	<22>\undefined organization\
							6F	69	74	61	7A	69	6E	0C	0169F	P.ABX:	.ASCII	<12>\fixed-length\
							2D	64	65	78	69	66	73	06	016A7	P.ABY:	.ASCII	<6>\stream\
							6D	61	65	72	74	73	09	016B4	P.ABZ:	.ASCII	<9>\stream-CR\	
							2D	6D	61	65	72	74	73	09	016BB	P.ACA:	.ASCII	<9>\stream-LF\
66	20	64	65	6E	69	66	65	64	6E	75	20	6E	61	13	016C5	P.ACB:	.ASCII	<19>\an undefined format\
74	67	6E	65	6C	20	65	6C	62	61	69	72	61	76	0F	016DE	P.ACC:	.ASCII	<15>\variable length\
63	20	29	52	43	28	20	64	65	69	6C	70	6D	69	1D	016F3	P.ACD:	.ASCII	<29>\implied (CR) carriage control\
6C	6F	72	74	6E	6F	63	20	65	67	61	69	72	72	61	01702	P.ACE:	.ASCII	<30>\FORTRAN (FTN) carriage control\
20	29	4E	54	46	28	20	4E	41	52	54	52	4F	46	1E	01711	P.ACF:	.ASCII	\!print file (PRN) carriage control\
6F	72	74	6E	6F	63	20	65	67	61	69	72	72	61	63	01720	P.ACG:	.ASCII	\''imbedded (<none>) carriage control\
52	50	28	20	65	6C	69	66	20	74	6E	69	72	70	21	01730			
6E	6F	63	20	65	67	61	69	72	72	61	63	20	29	4E	0173F			
6E	6F	63	3C	28	20	64	65	64	64	65	62	6D	69	22	0174E			
6F	63	20	65	67	61	69	72	72	61	63	20	29	3E	65	01752			
															01761			
															01770			

FORMAT_POS=	P.ABH
RECORD_FORMAT=	P.ABJ
RECORD_VFC_FORMAT=	P.ABL
REC_SIZE=	P.ABN
CARRIAGE_FORMAT=	P.ABP
FORMAT_NEG=	P.ABR

07FC 00000 GET FILE DESCRIPTION:

CODE SECTION DESCRIPTION					
				.WORD	Save R2,R3,R4,R5,R6,R7,R8,R9,R10
5A	00000000G	00	9E 00002	MOVAB	SYSSFAO, R10
59	FEF2	CF	9E 00009	MOVAB	P.ABT, R9
5E		10	C2 0000E	SUBL2	#16. \$P
50	04	AC	7D 00011	MOVQ	SCB, R0
52	0248	CO	D0 00015	MOVL	584(R0), R2
54	024C	CO	D0 0001A	MOVL	588(R0), R4
57	0254	CO	D0 0001F	MOVL	596(R0), R7
53	0258	CO	D0 00024	MOVL	600(R0), R3
56	025C	CO	D0 00029	MOVL	604(R0), R6
		7E	D4 0002E	CLRL	CURRENT LEN
04	AE 0200	8F	3C 00030	MOVZWL	#512, STRING_PTR
08	AE 04	A1	D0 00036	MOVL	4(R1), STRING_PTR+4
	55 0C	AC	D0 0003B	MOVL	RET LEN, R5
		65	B4 0003F	CLRW	(R5)
58	0098	CO	9E 00041	MOVAB	152(R0), R8
		50	DD 00046	PUSHL	R0
0000V	CF	01	FB 00048	CALLS	#1, FILE_OPEN
03		50	F8 0004D	BLBS	R0 1\$
	0162	31	00050	BRW	19\$
51	10	A3	D0 00053	MOVL	16(R3), FILE_SIZE
	14	A3	B5 00057	TSTW	20(R3)
		06	12 0005A	BNEQ	2\$
		51	D5 0005C	TSTL	FILE_SIZE
		02	13 0005E	BEQL	2\$

SEPARATE
V04-001

Print Symbiont -- separation routines 16-Sep-1984 02:23:03 VAX-11 BLISS-32 V4.0-742
GET FILE DESCRIPTION - Create a Sentence Descri 14-Sep-1984 22:32:26 [PRTSMB.SRC]SEPARATE.B32:2

Page 100
(26)

20	1D	51	D7	00060	2\$:	DECL	FILE SIZE
		A2	91	00062		CMPB	29(R2), #32
50		05	12	00066		BNEQ	3\$
		69	9E	00068		MOVAB	P.ABT, ORGANIZATION
	1D	1B	11	0006B	3\$:	BRB	6\$
50	08	A2	95	0006D		TSTB	29(R2)
50	06	12	00070			BNEQ	4\$
50	08	A9	9E	00072		MOVAB	P.ABU, ORGANIZATION
10	1D	10	11	00076		BRB	6\$
10	06	A2	91	00078	4\$:	CMPB	29(R2), #16
50	13	A9	9E	0007C		BNEQ	5\$
50	04	9E	00082			MOVAB	P.ABV, ORGANIZATION
50	1C	A9	9E	00084	5\$:	BRB	6\$
	0C	A6	DD	00088	6\$:	MOVAB	P.ABW, ORGANIZATION
		50	DD	0008B		PUSHL	12(R6\$)
		51	DD	0008D		PUSHL	ORGANIZATION
	0C	A7	9F	0008F		PUSHL	FILE SIZE
7E	28	A4	3C	00092		PUSHAB	12(R7)
7E	26	A4	3C	00096		MOVZWL	40(R4), -(SP)
7E	24	A4	3C	0009A		MOVZWL	38(R4), -(SP)
		58	DD	0009E		MOVZWL	36(R4), -(SP)
	24	AE	9F	000A0		PUSHL	R8
	24	AE	9F	000A3		PUSHAB	STRING_PTR
	FF04	C9	9F	000A6		PUSHAB	CURRENT_LEN
	6A	0B	FB	000AA		PUSHAB	FORMAT_POS
	65	6E	A0	000AD		CALLS	#11, \$\$\$\$FA0
	AE	6E	C0	000B0		ADDW2	CURRENT_LEN, (R5)
	AE	65	3C	000B4		ADDL2	CURRENT_LEN, STRING_PTR+4
	BF	04	AE	C3	000B8	MOVZWL	(R5), STRING_PTR
	54	0A	A3	3C	000C2	SUBL3	STRING_PTR, #512, STRING_PTR
	53	1F	A2	9A	000C6	MOVZWL	10(R3), RECORD_SIZE
	03	53	91	000CA		MOVZBL	31(R2), R3
		4D	13	000CD		CMPB	R3, #3
	01	53	91	000CF		BEQL	13\$
		4D	13	000CD		CMPB	R3, #1
	01	53	91	000CF		BNEQ	7\$
	50	33	A9	9E	000D4	MOVAB	P.ABX, FORMAT
		34	11	000D8		BRB	12\$
	04	53	91	000DA	7\$:	CMPB	R3, #4
		06	12	000DD		BNEQ	8\$
	50	40	A9	9E	000DF	MOVAB	P.ABY, FORMAT
		29	11	000E3		BRB	12\$
	06	53	91	000E5	8\$:	CMPB	R3, #6
	50	47	A9	9E	000EA	BNEQ	9\$
		1E	11	000EE		MOVAB	P.ABZ, FORMAT
	05	53	91	000FO	9\$:	BRB	12\$
		06	12	000F3		CMPB	R3, #5
	50	51	A9	9E	000F5	BNEQ	10\$
		13	11	000F9		MOVAB	P.ACA, FORMAT
		53	D5	000FB	10\$:	BRB	12\$
	50	58	A9	9E	000FF	TSTL	R3
		06	12	000FD		BNEQ	11\$
	02	53	91	00103		MOVAB	P.ACB, FORMAT
		09	11	00105	11\$:	BRB	12\$
	50	6F	A9	9E	0010A	CMPB	R3, #2
		04	12	00108		BNEQ	12\$
	50					MOVAB	P.ACC, FORMAT

SEPARATE
V04-001

Print_Symbiont -- separation routines
GET_FILE_DESCRIPTION - Create a Senter

B 5
16-Sep-1984 02:23:03
14-Sep-1984 22:32:26

VAX-11 Bliss-32 V4.0-742
[PRTSMB.SRC]SEPARATE.B32:2

Page 101
(26)

; Routine Size: 467 bytes, Routine Base: CODE + 1775

SEPARATE
V04-001

Print Symbiont -- separation routines
GET_FILE_DESCRIPTION - Create a Sentence Descri

C 5
16-Sep-1984 02:23:03
14-Sep-1984 22:32:26

VAX-11 Bliss-32 V4.0-742
[PRTSMB.SRC]SEPARATE.B32;2

Page 102
(26)

SE
VO
:

```

: 3096    4017 1 %sbttl 'GET_FILE_NAME - Get Name of the Current File'
: 3097    4018 1 ++
: 3098    4019 1 Functional Description:
: 3099    4020 1 This routine creates a phrase with the name of the current file.
: 3100    4021 1
: 3101    4022 1 Formal Parameters:
: 3102    4023 1      SCB          - Address of the SCB
: 3103    4024 1      STR_DESC    - Desc of String to Return
: 3104    4025 1      RET_LEN     - Return Length of Desc.
: 3105    4026 1
: 3106    4027 1 Implicit Inputs:
: 3107    4028 1      none
: 3108    4029 1
: 3109    4030 1 Implicit Outputs:
: 3110    4031 1      none
: 3111    4032 1
: 3112    4033 1 Returned Value:
: 3113    4034 1      none
: 3114    4035 1
: 3115    4036 1 Side Effects:
: 3116    4037 1      none
: 3117    4038 1      --
: 3118    4039 1 ROUTINE GET_FILE_NAME (
: 3119    4040 1      SCB          : REF $BLOCK,
: 3120    4041 1      EXPECTED_LEN   : REF VECTOR[2],
: 3121    4042 1      STR_DESC    : REF VECTOR[2],
: 3122    4043 1      RET_LEN     : REF VECTOR [,WORD]
: 3123    4044 1              ) : NOVALUE =
: 3124    4045 2 BEGIN
: 3125    4046 2 BIND
: P 4047 2      SENT_FORMAT = $descriptor (
: 3126    4048 2      "AS");
: 3127    4049 2
: 3128    4050 2 LOCAL
: 3129    4051 2      BUFFER      : VECTOR [512,byte],
: 3130    4052 2      LENG        : VECTOR [1],
: 3131    4053 2      NAME         : VECTOR[2];
: 3132    4054 2
: 3133    4055 2      NAME[SIZE] = %ALLOCATION(BUFFER);      | allocate for routines
: 3134    4056 2      NAME[ADDR] = BUFFER;                  | init address
: 3135    4057 2
: P 4058 2      $FAO (  SENT_FORMAT,
: P 4059 2      NAME[SIZE],                                | return length
: P 4060 2      NAME,                                    | address of string
: P 4061 2      SCB[PMSMQ_FILE_SPECIFICATION],       | file name
: 3140    4062 2      );
: 3141    4063 2
: 3142    4064 2      LENG[0] = .EXPECTED_LEN;           ! must be reference to word for call
: 3143    4065 2
: 3144    4066 2      IF .RET_LEN[0] GTR .EXPECTED_LEN THEN
: 3145    4067 2          Trim the file spec to fit.
: 3146    4068 2
: 3147    4069 2          LIBSTRIM_FILESPEC ( NAME, STR_DESC[0], LENG[0],
: 3148    4070 2                      STR_DESC[SIZE])
: 3149    4071 2
: 3150    4072 2      ELSE
: 3151    4073 3          BEGIN
: 3152    4073 3          STR_DESC[SIZE] = .NAME[SIZE];

```

SEPARATE
VO4-001

Print Symbiont -- separation routines
GET_FILE_NAME - Get Name of the Current File

E 5
16-Sep-1984 02:23:03
14-Sep-1984 22:32:26

VAX-11 Bliss-32 V4.0-742
[PRTSMB.SRC]SEPARATE.B32;2

Page 104
(27)

: 3153 4074 3 STR_DESC[ADDR] = .NAME[ADDR];
: 3154 4075 2 END:
: 3155 4076 2
: 3156 4077 2 RET_LEN[0] = .STR_DESC[SIZE];
: 3157 4078 2
: 3158 4079 2 RETURN SSS_NORMAL;
: 3159 4080 1 END:

53 41 21 01948 P.AC1: .ASCII \!AS\
0194B .BLKB 1
00000003 0194C P.ACH: .LONG 3
00000000 01950 .ADDRESS P.AC1

SENT_FORMAT= P.ACH

0004 00000 GET_FILE_NAME:
.WORD Save R2 4039
04 SE FDF4 CE 9E 00002 MOVAB -524(SP), SP
08 AE 0200 8F 3C 00007 MOVZWL #512, NAME 4055
7E 04 AC 00000098 AE 9E 0000D MOVAB BUFFER, NAME+4 4056
08 0C 00000008 C1 00012 ADDL3 #152, SCB, -(SP) 4062
08 AE 9F 0001B PUSHAB NAME
0C AE 9F 0001E PUSHAB NAME
D4 AF 9F 00021 PUSHAB SENT FORMAT
00000000G 00 04 FB 00024 CALLS #4, SYSSFAO
6E 08 AC D0 0002B MOVL EXPECTED_LEN, LEN 4064
08 0C AC D0 0002F MOVL STR_DESC, R2 4070
08 52 00033 CMPZV #0, #16, @RET_LEN, EXPECTED_LEN 4066
08 10 13 15 0003A BLEQ 1\$ 4070
08 52 DD 0003C PUSHL R2 4069
08 04 AE 9F 0003E PUSHAB LENGTH 4070
08 52 DD 00041 PUSHL R2 4069
08 10 AE 9F 00043 PUSHAB NAME 4070
00000000G 00 04 FB 00046 CALLS #4, LIBSTRIM_FILESPEC 4070
10 62 04 04 11 0004D BRB 2\$ 4073
10 BC 04 AE 7D 0004F 1\$: MOVQ NAME, (R2)
10 62 B0 00053 2\$: MOVW (R2), @RET_LEN 4077
10 04 00057 RET 4080

; Routine Size: 88 bytes. Routine Base: CODE + 1954

3161 4081 1 %sbttl 'INSERT_FILENAME_BANNER - Get Name of the Current File'
3162 4082 1 ++
3163 4083 1 Functional Description:
3164 4084 1 This routine creates a banner phrase with the name of the current file.
3165 4085 1 Algorithm:
3166 4086 1 If the Filename, Type, and Version (FTV) fits on one line
3167 4087 1 If only one banner line exists...
3168 4088 1 insert FTV on only ONE line
3169 4089 1
3170 4090 1 If there are Three banner Lines available...
3171 4091 1 insert Filename on one, Type on
3172 4092 1 another, and Version on third
3173 4093 1
3174 4094 1 Otherwise...
3175 4095 1 insert Filename on one, Type and
3176 4096 1 Version on the second
3177 4097 1 Formal Parameters:
3178 4098 1 SCB : Address of the SCB
3179 4099 1 STR_DESC : Desc of String to Return
3180 4100 1 RET_LEN : Return Length of Desc.
3181 4101 1
3182 4102 1 Implicit Inputs:
3183 4103 1 none
3184 4104 1
3185 4105 1 Implicit Outputs:
3186 4106 1 none
3187 4107 1
3188 4108 1 Returned Value:
3189 4109 1 none
3190 4110 1
3191 4111 1 Side Effects:
3192 4112 1 none
3193 4113 1 --
3194 4114 1 ROUTINE INSERT_FILENAME_BANNER {
3195 4115 1 SCB : REF \$BBLOCK,
3196 4116 1 STR_DESC : REF VECTOR[2],
3197 4117 1 FRAME_PTR : REF PAGE_ARRAY,
3198 4118 1 FRAME_WIDTH , : Number of Columns
3199 4119 1 FRAME_LENGTH , : Number of Rows
3200 4120 1) =
3201 4121 2 BEGIN
3202 4122 2 LITERAL
3203 4123 2 BIG_BANNER = 14,
3204 4124 2 LITTLE_BANNER = 7,
3205 4125 2 SMALL = 2,
3206 4126 2 LARGE = 1;
3207 4127 2
3208 4128 2 LOCAL
3209 4129 2 RET_LEN : VECTOR[1],
3210 4130 2 PAGE_PTR : REF PAGE_ARRAY,
3211 4131 2 SPACING
3212 4132 2 CURRENT_PTR ,
3213 4133 2 FTV_LEN : VECTOR[1],
3214 4134 2 BANNER_TYPE ,
3215 4135 2 BANNER_SIZE ,
3216 4136 2 MAX_BAN_CHARS,
3217 4137 2 MAX_ROWS ,

```
: 3218      4138 2      MAX_COLS
: 3219      4139 2      NAME      : VECTOR[2];
: 3220      4140 2      TYPE      : VECTOR[2];
: 3221      4141 2      VERS      : VECTOR[2];
: 3222      4142 2
: 3223      4143 2      ! dont even try if there is no frame left
: 3224      4144 2
: 3225      4145 2      IF (.FRAME_LENGTH LSS 7)
: 3226      4146 2      THEN
: 3227      4147 2          RETURN 0;
: 3228      4148 2
: 3229      4149 2      PARSE_FILE_NAME (SCB[PSMSQ_FILE_SPECIFICATION], FSCNS_NAME, NAME);
: 3230      4150 2      PARSE_FILE_NAME (SCB[PSMSQ_FILE_SPECIFICATION], FSCNS_TYPE, TYPE);
: 3231      4151 2      PARSE_FILE_NAME (SCB[PSMSQ_FILE_SPECIFICATION], FSCNS_VERSION, VERS);
: 3232      4152 2
: 3233      4153 2      FTV_LEN[0] = .NAME[SIZE] + .TYPE[SIZE] + .VERS[SIZE];
: 3234      4154 2
: 3235      4155 2      BANNER_TYPE = BIG_BANNER;
: 3236      4156 2      BANNER_SIZE = LARGE;
: 3237      4157 2      SPACING = SMALL;                                ! two spaces between banner rows
: 3238      4158 2
: 3239      4159 2      MAX_BAN_CHARS = (.FRAME_LENGTH/16) * (.FRAME_WIDTH/12);
: 3240      4160 2      MAX_ROWS = .BANNER_SIZE * (.FRAME_LENGTH / 18);
: 3241      4161 2      MAX_COLS = .BANNER_SIZE * (.FRAME_WIDTH / 12);
: 3242      4162 2
: 3243      4163 2      IF .MAX_BAN_CHARS LSS .FTV_LEN[0]
: 3244      4164 2      THEN
: 3245      4165 3      BEGIN
: 3246      4166 3      BANNER_SIZE = SMALL;
: 3247      4167 3      BANNER_TYPE = LITTLE_BANNER;
: 3248      4168 3      SPACING = LARGE;                                ! single space banner rows
: 3249      4169 3      MAX_ROWS = .BANNER_SIZE * (.FRAME_LENGTH / 18);
: 3250      4170 3      MAX_COLS = .BANNER_SIZE * (.FRAME_WIDTH / 14);
: 3251      4171 2      END;
: 3252      4172 2
: 3253      4173 2
: 3254      4174 2      ! Do somemore calculations to ensure consistent letter sizing
: 3255      4175 2      !X! Just to gte this out the door... needs to be optimized later.rb
: 3256      4176 2
: 3257      4177 4      IF ( (.NAME[SIZE] GTR .MAX_COLS)
: 3258                  OR
: 3259                  (.TYPE[SIZE] GTR .MAX_COLS)
: 3260                  OR
: 3261                  (.VERS[SIZE] GTR .MAX_COLS)
: 3262                  OR
: 3263                  (.MAX_ROWS LSS 3) AND
: 3264                  ((.TYPE[SIZE] + .VERS[SIZE]) GTR .MAX_COLS)
: 3265                  OR
: 3266                  (.FTV_LEN[0] GTR (.MAX_COLS * .MAX_ROWS)) )
: 3267      4187 3
: 3268      4188 2      THEN
: 3269      4189 3      BEGIN
: 3270      4190 3      BANNER_SIZE = SMALL;
: 3271      4191 3      BANNER_TYPE = LITTLE_BANNER;
: 3272      4192 3      SPACING = LARGE;                                ! single space banner rows
: 3273      4193 3      MAX_ROWS = .BANNER_SIZE * (.FRAME_LENGTH / 18);
: 3274      4194 3      MAX_COLS = .BANNER_SIZE * (.FRAME_WIDTH / 14);
```

```
3275 4195 2      END;
3276 4196 2
3277 4197 2      ! Attempt to fit the filename, type, and version on one line
3278 4198 2
3279 4199 2      IF (.FTV_LEN[0] LEQ .MAX_COLS)           ! insert on one line
3280 4200 2          AND                                ! only if little banner
3281 4201 2          (.BANNER_TYPE EQL LITTLE_BANNER)
3282 4202 2
3283 4203 2      THEN
3284 4204 2          BEGIN
3285 4205 2              CURRENT_PTR = .STR_DESC[ADDR];
3286 4206 2              CURRENT_PTR = CHSMOVE(.NAME[SIZE], .NAME[ADDR], .CURRENT_PTR);
3287 4207 2              CURRENT_PTR = CHSMOVE(.TYPE[SIZE], .TYPE[ADDR], .CURRENT_PTR);
3288 4208 2              CURRENT_PTR = CHSMOVE(.VERS[SIZE], .VERS[ADDR], .CURRENT_PTR);
3289 4209 2              STR_DESC[SIZE] = .FTV_LEN[0];
3290 4210 2
3291 4211 2      INSERT_NAME_BANNER (.SCB,
3292 4212 2          STR_DESC[SIZE], ! job name desc
3293 4213 2          FRAME_PTR[0,0..SCB[PSMSL PAGE_WIDTH]], ! ref-to frame
3294 4214 2          .FRAME_WIDTH, ! max width Bann
3295 4215 2          .BANNER_TYPE, ! frame length
3296 4216 2          .BANNER_TYPES); ! max hght Bann str
3297 4217 2
3298 4218 2      RETURN .BANNER_TYPE; ! return how much space used
3299 4219 2
3300 4220 2      END
3301 4221 2
3302 4222 2      BEGIN
3303 4223 2          ! Move filename with truncated banners when not enough space
3304 4224 2          IF (      (.MAX_ROWS LEQ 1) )
3305 4225 2
3306 4226 2          !X! Comment this out .... causes too many filenames to be printed on
3307 4227 2          !X! a single line when two lines would be more appropriate.
3308 4228 2
3309 4229 2          OR
3310 4230 2          (.NAME[SIZE] GTR .MAX_COLS)
3311 4231 2          OR
3312 4232 2          (.TYPE[SIZE] GTR .MAX_COLS)
3313 4233 2          OR
3314 4234 2          (.VERS[SIZE] GTR .MAX_COLS)
3315 4235 2          OR
3316 4236 2          (      (.MAX_ROWS LSS 3) AND
3317 4237 2          ((.TYPE[SIZE] + .VERS[SIZE]) GTR .MAX_COLS)
3318 4238 2          OR
3319 4239 2          (.FTV_LEN[0] GTR (.MAX_COLS * .MAX_ROWS)) )
3320 4240 2
3321 4241 2
3322 4242 2      THEN
3323 4243 2          BEGIN
3324 4244 2              CURRENT_PTR = .STR_DESC[ADDR];
3325 4245 2              CURRENT_PTR = CHSMOVE(.NAME[SIZE], .NAME[ADDR], .CURRENT_PTR);
3326 4246 2              CURRENT_PTR = CHSMOVE(.TYPE[SIZE], .TYPE[ADDR], .CURRENT_PTR);
3327 4247 2              CURRENT_PTR = CHSMOVE(.VERS[SIZE], .VERS[ADDR], .CURRENT_PTR);
3328 4248 2
3329 4249 2          !X! This is a cludge to get this out the door. I will declare a valid
3330 4250 2          descriptor in the future and use LIB$TRIM_FILESPEC.
3331 4251 2
3332 4252 2      STR_DESC[SIZE] = .FTV_LEN[0];
```

```
3332      4252 4
3333      4253 4
3334      4254 4
3335      4255 5
3336      4256 4
3337      4257 5
3338      4258 5
3339      4259 5
3340      4260 5
3341      4261 6
3342      4262 5
3343      4263 4
3344      4264 4
3345      4265 4
3346      4266 4
3347      4267 4
3348      4268 4
3349      4269 4
3350      4270 4
3351      4271 4
3352      4272 4
3353      4273 4
3354      4274 4
3355      4275 4
3356      4276 4
3357      4277 4
3358      4278 4
3359      4279 4
3360      4280 4
3361      4281 4
3362      4282 4
3363      4283 4
3364      4284 4
3365      4285 3
3366      4286 4
3367      4287 4
3368      4288 4
3369      4289 5
3370      4290 5
3371      4291 5
3372      4292 5
3373      4293 5
3374      4294 5
3375      4295 5
3376      4296 5
3377      4297 5
3378      4298 5
3379      4299 5
3380      4300 5
3381      4301 5
3382      4302 5
3383      4303 5
3384      4304 5
3385      4305 5
3386      4306 5
3387      4307 5
3388      4308 5

        IF .NAME[0] GEQ .MAX_COLS
        THEN
            (STR_DESC[SIZE] = .MAX_COLS)
        ELSE
            (IF .NAME[0]+.TYPE[0] GTR .MAX_COLS
            THEN
                STR_DESC[SIZE] = .NAME[0]
            ELSE
                IF ((.NAME[0]+TYPE[0]+VERS[0]) GTR .MAX_COLS)
                THEN
                    STR_DESC[SIZE] = .NAME[0]+.TYPE[0]);
            IF .FTV_LEN[0] GTR (.MAX_ROWS * .MAX_COLS)
            THEN
                ! Trim the file spec to fit.
                LIB$TRIM_FILESPEC ( STR_DESC, STR_DESC[0],
                                     .MAX_COLS,
                                     STR_DESC[SIZE]);
        RET_LEN[0] = INSERT_NAME_BANNER (
            .SCB,
            STR_DESC[SIZE], ! job name desc
            FRAME_PTR[0,0,.SCB[PSMSL_PAGE_WIDTH]],
            .FRAME_WIDTH, ! ref to frame
            .FRAME_WIDTH, ! max width Bann
            .BANNER_TYPE, ! frame length
            .BANNER_TYPE); ! max hght Bann str
        RETURN .RET_LEN[0]; ! return how much space used
    END
    ELSE ! Should be able to insert it... Make it pretty
    BEGIN
        IF .MAX_ROWS GEQ 3
        THEN
            BEGIN
                PAGE_PTP = FRAME_PTR[0,0,.SCB[PSMSL_PAGE_WIDTH]];
                INSERT_NAME_BANNER (
                    .SCB,
                    NAME[SIZE],
                    PAGE_PTR[0,0,.SCB[PSMSL_PAGE_WIDTH]], ! file name
                    .FRAME_WIDTH, ! ref to frame
                    .FRAME_WIDTH, ! max width Bann
                    .BANNER_TYPE, ! frame length
                    .BANNER_TYPE); ! max hght Bann str
                PAGE_PTR = PAGE_PTR[0,(.BANNER_TYPE+.SPACING),
                                     .SCB[PSMSL_PAGE_WIDTH]];
                INSERT_NAME_BANNER (
                    .SCB,
                    TYPE[SIZE],
                    PAGE_PTR[0,0,.SCB[PSMSL_PAGE_WIDTH]], ! file type
                    .FRAME_WIDTH, ! ref to frame
                    .FRAME_WIDTH, ! max width Bann

```

SEPARATE
V04-001

Print_Symbiont -- separation routines
INSERT_FILENAME_BANNER - Get Name of

J 5
16-Sep-1984 02:23:03
14-Sep-1984 22:32:26

VAX-11 Bliss-32 V4.0-742
[PRTSMB.SRC]SEPARATE.B32;2

Page 109
(28)

SE
VC

OFFC 00000 INSERT_FILENAME_BANNER:

SEPARATE
V04-001

Print Symbiont -- separation routines
INSERT_FILENAME BANNER - Get Name of the

K 5
16-Sep-1984 02:23:03
14-Sep-1984 22:32:26

VAX-11 Bliss-32 V4.0-742
[PRTSMB.SRC]SEPARATE.B32:2

Page 110
(28)

SB
VO

					.WORD	Save R2,R3,R4,R5,R6,R7,R8,R9,R10,R11	4114
					SUBL2	#32, SP	4145
					CMPL	FRAME_LENGTH, #7	
					BGEQ	1\$	
					BRW	12\$	
				18:	PUSHAB	NAME	
					PUSHL	#6	
					ADDL3	#152, SCB, R0	
					MOVAB	(R0), R2	
					PUSHL	R2	
					CALLS	#3, PARSE_FILE_NAME	
					PUSHAB	TYPE	
				10:	PUSHL	#7	
					PUSHL	R2	
					CALLS	#3, PARSE_FILE_NAME	
					PUSHAB	VERS	
					PUSHL	#8	
					PUSHL	R2	
					CALLS	#3, PARSE_FILE_NAME	
					MOVL	NAME, R11	
					MOVAB	@TYPE[R11], 4(SP)	
					ADDL3	VERS, 4(SP), FTV_LEN	
					MOVL	#14, BANNER_TYPE	
					MOVL	#1, BANNER_SIZE	
					MOVL	#2, SPACING	
					DIVL3	#16, FRAME_LENGTH, R2	
					MOVL	FRAME_WIDTH, R10	
					DIVL3	#12, R10, R1	
					MULL3	R1, R2, MAX_BAN_CHARS	
					MULL3	R2, BANNER_SIZE, MAX_ROWS	
					MULL3	#1, BANNER_SIZE, MAX_COLS	
					CMPL	MAX_BAN_CHARS, FTV_LEN	
					BGEQ	2\$	
					MOVL	#2, BANNER_SIZE	
					MOVL	#7, BANNER_TYPE	
					MOVL	#1, SPACING	
					DIVL3	#18, FRAME_LENGTH, R1	
					MULL3	BANNER_SIZE, R1, MAX_ROWS	
					DIVL3	#14, RTO, R1	
					MULL3	BANNER_SIZE, R1, MAX_COLS	
					CMPL	R11, MAX_COLS	
					4\$		
					BGTR	TYPE, MAX_COLS	
					CMPL	4\$	
					BGTR	VERS, MAX_COLS	
					CMPL	4\$	
					BGEQ	MAX_ROWS, #3	
					ADDL3	3\$	
					CMPL	VERS, TYPE, R1	
					51	R1, MAX_COLS	
					09:	4\$	
					BGTR	MAX_ROWS, MAX_COLS, R1	
					CMPL	FTV_LEN, R1	
					BLEQ	5\$	
					MOVL	#2, BANNER_SIZE	
					MOVL	#7, BANNER_TYPE	
					MOVL	#1, SPACING	

SEPARATE
V04-001Print Symbiont -- separation routines
INSERT_FILENAME_BANNER - Get Name of the Curren

L

5

16-Sep-1984 14-Sep-1984 02:23:03 22:32:26

VAX-11 Bliss-32 v4.0-742
[PRTSMB.SRC]SEPARATE.B32;2Page 111
(28)

51	14	AC	12	C7	000C2	DIVL3	#18, FRAME_LENGTH, R1	: 4193		
59	51	50	C5	000C7	MULL3	BANNER_SIZE, R1, MAX_ROWS	: 4194			
51	5A	0E	C7	000CB	DIVL3	#14, RT0, R1	: 4195			
56	51	50	C5	000CF	MULL3	BANNER_SIZE, R1, MAX_COLS	: 4196			
	56	6E	D1	000D3	5\$: CMPL	FTV_LEN, MAX_COLS	: 4197			
	07	38	14	000D6	BGTR	6\$: 4198			
		33	12	000DB	CMPL	BANNER_TYPE, #7	: 4199			
		08	AC	000DD	BNEQ	6\$: 4200			
		58	A8	000E1	MOVL	STR DESC, R8	: 4201			
63	1C	BE	5B	28	000E5	MOVL	4(R8), CURRENT_PTR	: 4202		
63	14	BE	AE	28	000EA	MOV3	R11, &NAME+4, (CURRENT_PTR)	: 4203		
63	0C	BE	AE	28	000FO	MOV3	TYPE, &TYPE+4, (CURRENT_PTR)	: 4204		
	68	6E	DO	000F6	MOV3	VERS, &VERS+4, (CURRENT_PTR)	: 4205			
		57	DD	000F9	MOVL	FTV_LEN, (R8)	: 4206			
		57	DD	000FB	PUSHL	BANNER_TYPE	: 4207			
		5A	DD	000FD	PUSHL	BANNER_TYPE	: 4208			
		0C	AC	000FF	PUSHL	R10	: 4209			
		58	DD	00102	PUSHL	FRAME_PTR	: 4210			
		04	AC	00104	PUSHL	R8	: 4211			
0000V	CF	06	FB	00107	PUSHL	SCB	: 4212			
	50	57	DO	0010C	CALLS	#6, INSERT_NAME_BANNER	: 4213			
		04	0010F	MOVL	RET	BANNER_TYPE, R0	: 4214			
		01	59	D1	00110	6\$: CMPL	MAX_ROWS, #1	: 4215		
			5A	14	00113	BGTR	10\$: 4216		
		58	08	AC	00115	MOVL	STR DESC, R8	: 4217		
63	1C	BE	53	04	A8	MOVL	4(R8), CURRENT_PTR	: 4218		
63	14	BE	5B	28	0011D	MOV3	R11, &NAME+4, (CURRENT_PTR)	: 4219		
63	0C	BE	10	AE	28	MOV3	TYPE, &TYPE+4, (CURRENT_PTR)	: 4220		
		08	AE	28	00122	MOV3	VERS, &VERS+4, (CURRENT_PTR)	: 4221		
		68	6E	DO	0012E	MOVL	FTV_LEN, (R8)	: 4222		
		56	5B	D1	00131	CMPL	R11, MAX_COLS	: 4223		
			05	19	00134	BLSS	7\$: 4224		
		68	56	DO	00136	MOVL	MAX_COLS, (R8)	: 4225		
			20	11	00139	BRB	9\$: 4226		
		56	04	AE	D1	0013B	7\$: CMPL	4(SP), MAX_COLS	: 4227	
			05	15	0013F	BLEQ	8\$: 4228		
		68	5B	DO	00141	MOVL	R11, (R8)	: 4229		
			15	11	00144	BRB	9\$: 4230		
		50	10	AE	9E	00146	8\$: MOVAB	TYPE, R0	: 4231	
		50	08	AE	40	9E	0014A	MOVAB	VERS[R0], R0	: 4232
		50			5B	C0	0014F	ADDL2	R11, R0	: 4233
		56			50	D1	00152	CMPL	R0, MAX_COLS	: 4234
		68	04	AE	DO	00155	BLEQ	9\$: 4235	
			57	DD	00158	9\$: MOVL	4(SP), (R8)	: 4236		
			57	DD	0015D	PUSHL	BANNER_TYPE	: 4237		
			5A	DD	0015F	PUSHL	BANNER_TYPE	: 4238		
			0C	AC	DD	00161	PUSHL	R10	: 4239	
			58	DD	00164	PUSHL	FRAME_PTR	: 4240		
			04	AC	DD	00166	PUSHL	R8	: 4241	
0000V	CF	06	FB	00169	PUSHL	SCB	: 4242			
			04	0016E	CALLS	#6, INSERT_NAME_BANNER	: 4243			
			56	OC	AC	DO	0016F	10\$: RET	: 4244	
58	57	54	C1	00173	MOVL	FRAME_PTR, PAGE_PTR	: 4245			
50	04	AC	00000200	8F	C1	00177	ADDL3	SPACING, BANNER_TYPE, R8	: 4246	
		58		60	9E	00180	ADDL3	#512, SCB, R0	: 4247	
							MOVAB	(R0), R11	: 4248	

SE
VC

SEPARATE
V04-001Print Symbiont -- separation routines
INSERT_FILENAME_BANNER - Get Name of the Curren

M 5

16-Sep-1984 14-Sep-1984 02:23:03 22:32:26

VAX-11 Bliss-32 V4.0-742
[PRTSMB.SRC]SEPARATE.B32;2Page 112
(28)

			03	59 D1 00183	CMPL	MAX_ROWS, #3	: 4287
				4C 19 00186	BLSS	11\$: 4299
				57 DD 00188	PUSHL	BANNER_TYPE	: 4298
				57 DD 0018A	PUSHL	BANNER_TYPE	: 4295
			0440	8F BB 0018C	PUSHR	#^M<R6,R10>	: 4294
			28	AE 9F 00190	PUSHAB	NAME	: 4295
			04	AC DD 00193	PUSHL	SCB	: 4295
50	0000V	CF		06 FB 00196	CALLS	#6, INSERT_NAME_BANNER	: 4302
		58		68 C5 0019B	MULL3	(R11), R8, R0	: 4310
		56		50 CO 0019F	ADDL2	RO, PAGE_PTR	: 4309
				57 DD 001A2	PUSHL	BANNER_TYPE	: 4306
			0440	8F BB 001A4	PUSHR	#^M<R6,R10>	: 4305
			20	AE 9F 001AA	PUSHAB	TYPE	: 4306
			04	AC DD 001AD	PUSHL	SCB	: 4306
50	0000V	CF		06 FB 001B0	CALLS	#6, INSERT_NAME_BANNER	: 4313
		58		68 C5 001B5	MULL3	(R11), R8, R0	: 4313
		56		50 CO 001B9	ADDL2	RO, PAGE_PTR	: 4321
				57 DD 001BC	PUSHL	BANNER_TYPE	: 4320
			0440	8F BB 001C0	PUSHR	#^M<R6,R10>	: 4317
			18	AE 9F 001C4	PUSHAB	VERS	: 4316
			04	AC DD 001C7	PUSHL	SCB	: 4317
50	0000V	CF		06 FB 001CA	CALLS	#6, INSERT_NAME_BANNER	: 4323
		58		03 C5 001CF	MULL3	#3, R8, R0	: 4327
				04 001D3	RET		: 4328
			59	08 AC 001D4	11\$:	MOVL STR DESC, R9	: 4328
63	14	BE	53	04 A9 001D8	MOVL	4(R9), CURRENT_PTR	: 4329
63	OC	BE	08	AE 28 001DC	MOVC3	TYPE, @TYPE+4,-(CURRENT_PTR)	: 4330
69	10	AE	08	AE 28 001E2	MOVC3	VERS, @VERS+4,-(CURRENT_PTR)	: 4331
				AE C1 001E8	ADDL3	VERS, TYPE, (R9)	: 4341
				57 DD 001EE	PUSHL	BANNER_TYPE	: 4340
				57 DD 001F0	PUSHL	BANNER_TYPE	: 4337
			0440	8F BB 001F2	PUSHR	#^M<R6,R10>	: 4336
			28	AE 9F 001F6	PUSHAB	NAME	: 4337
			04	AC DD 001F9	PUSHL	SCB	: 4337
50	0000V	CF		06 FB 001FC	CALLS	#6, INSERT_NAME_BANNER	: 4344
		58		68 C5 00201	MULL3	(R11), R8, R0	: 4344
		56		50 CO 00205	ADDL2	RO, PAGE_PTR	: 4352
				57 DD 00208	PUSHL	BANNER_TYPE	: 4351
			0440	8F BB 0020C	PUSHL	BANNER_TYPE	: 4348
			04	59 DD 00210	PUSHR	#^M<R6,R10>	: 4348
50	0000V	CF		04 AC DD 00212	PUSHL	R9	: 4354
		58		06 FB 00215	CALLS	#6, INSERT_NAME_BANNER	: 4221
				01 78 0021A	ASHL	#1, R8, R0	: 4359
				04 0021E	RET		: 4359
				50 D4 0021F	12\$:	CLRL	
				04 00221	RET	RO	

; Routine Size: 546 bytes, Routine Base: CODE + 19AC

SEPARATE
VO4-001

Print Symbiont -- separation routines
INSERT_JOBNUMBER_BANNER - Get Job Number of the current Job

N 5

16-Sep-1984 02:23:03
14-Sep-1984 22:32:26

VAX-11 Bliss-32 v4.0-742
[PRTSMB.SRC]SEPARATE.B32;2

Page 113
(29)

3441 4360 1 %sbttl 'INSERT_JOBNUMBER_BANNER - Get Job Number of the current Job'
3442 4361 1 ++
3443 4362 1 Functional Description:
3444 4363 1 This routine creates a banner phrase with the Job Number
3445 4364 1
3446 4365 1 Formal Parameters:
3447 4366 1 SCB - Address of the SCB
3448 4367 1 STR_DESC - Desc of String to Return
3449 4368 1 RET_LEN - Return Length of Desc.
3450 4369 1
3451 4370 1 Implicit Inputs:
3452 4371 1 none
3453 4372 1
3454 4373 1 Implicit Outputs:
3455 4374 1 none
3456 4375 1
3457 4376 1 Returned Value:
3458 4377 1 none
3459 4378 1
3460 4379 1 Side Effects:
3461 4380 1 none
3462 4381 1 --
3463 4382 1 ROUTINE INSERT_JOBNUMBER_BANNER (
3464 4383 1 SCB : REF \$BBLOCK,
3465 4384 1 STR_DESC : REF VECTOR[2],
3466 4385 1 FRAME_PTR : REF PAGE_ARRAY,
3467 4386 1 FRAME_WIDTH , | Number of Columns
3468 4387 1 FRAME_LENGTH , | Number of Rows
3469 4388 1) =
3470 4389 2 BEGIN
3471 4390 2 BIND
P 4391 2 SENT_FORMAT = \$DESCRIPTOR ('JOB'
P 4392 2 ' !UL');
P 4393 2
P 4395 2 NUM_FORMAT = \$DESCRIPTOR ('!UL');
3476 4396 2
3477 4397 2 LITERAL
3479 4398 2 LITTLE_BANNER = 7,
3480 4399 2 SMALL = 2,
3481 4400 2 LARGE = 1;
3482 4401 2
3483 4402 2
3484 4403 2 LOCAL
3485 4404 2 RET_LEN : VECTOR[1],
3486 4405 2 PAGE_PTR : REF PAGE_ARRAY,
3487 4406 2 CURRENT_PTR :
3488 4407 2 JOB_LEN : INITIAL (0),
3489 4408 2 BANNER_TYPE ,
3490 4409 2 SPACING ,
3491 4410 2 BANNER_SIZE ,
3492 4411 2 MAX_BAN_CHARS ,
3493 4412 2 BUFFER : VECTOR[10,byte],
3494 4413 2 MAX_COLS : VECTOR[2];
3495 4414 2 NUMBER : VECTOR[2];
3496 4415 2
3497 4416 2 NUMBER[SIZE] = %ALLOCATION(BUFFER);

```

: 3498
: 3499
: 3500
: 3501
: 3502
: 3503
: 3504
: 3505
: 3506
: 3507
: 3508
: 3509
: 3510
: 3511
: 3512
: 3513
: 3514
: 3515
: 3516
: 3517
: 3518
: 3519
: 3520
: 3521
: 3522
: 3523
: 3524
: 3525
: 3526
: 3527
: 3528
: 3529
: 3530
: 3531
: 3532
: 3533
: 3534
: 3535
: 3536
: 3537
: 3538
: 3539
: 3540
: 3541
: 3542
: 3543
: 3544
: 3545
: 3546
: 3547
: 3548
: 3549
: 3550
: 3551
: 3552
: 3553
: 3554
4417 2 NUMBER[ADDR] = BUFFER;
4418 2
4419 2 ! dont even try if there is no frame left
4420 2
4421 2 IF (.FRAME_LENGTH LSS 7)
4422 2 THEN
4423 2     RETURN 0;
4424 2
4425 2 MAX_BAN_CHARS = (.FRAME_LENGTH/16) * (.FRAME_WIDTH/12);
4426 2
4427 2 BANNER_SIZE = SMALL;
4428 2 BANNER_TYPE = LITTLE_BANNER;
4429 2 SPACING = LARGE;                                ! single space banner rows
4430 2
4431 2 MAX_COLS = .BANNER_SIZE * (.FRAME_WIDTH / 12);
4432 2
4433 2 P 4434 2 SFAD ( NUM_FORMAT,
4434 2             JOB_LEN,
4435 2             NUMBER[0],                                ! str[size] > fetched name len
4436 2             .SCB[PSMSL_ENTRY_NUMBER]
4437 2             );
4438 2
4439 2 IF (.JOB_LEN+4) LEQ .MAX_COLS                  ! insert job&num on one line
4440 2 THEN
4441 2 BEGIN
4442 3   P 4443 3 SFAD ( SENT_FORMAT,
4443 3             JOBLEN,
4444 3             STR_DESC[0],
4445 3             .SCB[PSMSL_ENTRY_NUMBER]                  ! job number
4446 3             );
4447 3
4448 3   STR_DESC[SIZE] = .JOB_LEN;                    ! update the size
4449 3 END
4450 2 ELSE
4451 3   BEGIN
4452 3     IF .JOB_LEN LEQ .MAX_COLS
4453 3     THEN
4454 4       BEGIN
4455 4         CURRENT_PTR = .STR_DESC[ADDR];
4456 4         CURRENT_PTR = CHSMOVE(.JOB_LEN, .NUMBER[ADDR], .CURRENT_PTR);
4457 4         STR_DESC[SIZE] = .JOB_LEN;
4458 4       END
4459 3     ELSE
4460 3       RETURN 0;                                ! exit ... No Room
4461 2   END;
4462 2
4463 2 INSERT_NAME_BANNER (
4464 2             .SCB,
4465 2             STR_DESC[SIZE], ! job name desc
4466 2             FRAME_PTR[0,0..SCB[PSMSL PAGE_WIDTH]], ! ref to frame
4467 2             .FRAME_WIDTH, ! max width Bann
4468 2             .BANNER_TYPE, ! frame length
4469 2             .BANNER_TYPE); ! max hght Bann str
4470 2
4471 2
4472 2 RETURN .BANNER_TYPE;                          ! return how much space in
4473 2                                         length used

```

SEPARATE
V04-001
: 3555

Print Symbiont -- separation routines
INSERT JOBNUMBER BANNER - Get Job Num

C 6
16-Sep-1984 02:23:03
14-Sep-1984 22:32:26

VAX-11 Bliss-32 V4.0-742
[PRTSMB.SRC]SEPARATE.B32;2

Page 115
(29)

Print Symbiont -- separation routines 16-Sep-1984 02:23:03 VAX-11 Bliss-32 v4.0-742
INSERT_JOBNUMBER_BANNER - Get Job Number of the 14-Sep-1984 22:32:26 [PRTSMB.SRC]SEPARATE.B32;2
4474 1 END:

03FC 00000 INSERT_JOBNUMBER BANNER:							
				WORD	Save R2,R3,R4,R5,R6,R7,R8,R9		
59	00000000G	00	9E 00002	MOVAB	SYSSFA0, R9		
SE		14	C2 00009	SUBL2	#20, SP		
04	AE	0C	DA DO 0000E	CLRL	JOB_LEN		
08	AE	0C	AE 9E 00012	MOVL	#10, NUMBER		
07		14	AC D1 00017	MOVAB	BUFFER, NUMBER+4		
			76 19 0001B	CMPL	FRAME_LENGTH, #7		
51	14	AC	10 C7 0001D	BLSS	3S		
50	10	AC	0C C7 00022	DIVL3	#16, FRAME_LENGTH, R1		
			50 C4 00027	DIVL3	#12, FRAME_WIDTH, R0		
51			02 DO 0002A	MULL2	R0, MAX_BAN_CHARS		
51			07 DO 0002D	MOVL	#2, BANNER_SIZE		
58			01 DO 00030	MOVL	#7, BANNER_TYPE		
52			50 C5 00033	MULL3	R0, BANNER_SIZE, MAX_COLS		
51	57	04	AC DO 00037	MOVL	SCB, R7		
		58	A7 DD 0003B	PUSHL	88(R7)		
		08	AE 9F 0003E	PUSHAB	NUMBER		
		08	AE 9F 00041	PUSHAB	JOB_LEN		
		B1	AF 9F 00044	PUSHAB	NUM_FORMAT		
50	69		04 FB 00047	CALLS	#4, SYSSFA0		
	6E		04 C1 0004A	ADDL3	#4, JOB_LEN, R0		
	52		50 D1 0004E	CMPL	R0, MAX_COLS		
			15 14 00051	BGTR	1S		
		58	A7 DD 00053	PUSHL	88(R7)		
		08	AC DD 00056	PUSHL	STR_DESC		
		08	AE 9F 00059	PUSHAB	JOB_LEN		
		8D	AF 9F 0005C	PUSHAB	SENT_FORMAT		
08	69		04 FB 0005F	CALLS	#4, SYSSFA0		
	BC		6E DO 00062	MOVL	JOB_LEN, @STR_DESC		
			15 11 00066	BRB	2S		
	52		6E D1 00068	CMPL	JOB_LEN, MAX_COLS		
			26 14 0006B	BGTR	3S		
56	08	AC	DO 0006D	MOVL	STR_DESC, R6		
53	04	A6	DO 00071	MOVL	4(R6), CURRENT_PTR		
08	BE	6E	28 00075	MOVC3	JOB_LEN, @NUMBER+4, (CURRENT_PTR)		
66		6E	DO 0007A	MOVL	JOB_LEN, (R6)		
63		58	DD 0007D	PUSHL	BANNER_TYPE		

SEPARATE
V04-001

Print Symbiont -- separation routines
INSERT_JOBNUMBER_BANNER - Get Job Number of the

D 6

16-Sep-1984 02:23:03

14-Sep-1984 22:32:26

VAX-11 Bliss-32 V4.0-742
[PRTSMB.SRC]SEPARATE.B32;2

Page 116
(29)

7E	OC	58	DD	0007F	PUSHL	BANNER_TYPE	4469
	08	AC	7D	00081	MOVQ	FRAME_PTR, -(SP)	4466
		AC	DD	00085	PUSHL	STR_DESC	
		57	DD	00088	PUSHL	R7	
0000V	CF	06	FB	0008A	CALLS	#6, INSERT_NAME_BANNER	
	50	58	DD	0008F	MOVL	BANNER_TYPE, R0	4472
		04	00092		RET		
		50	D4	00093 38:	CLRL	R0	4474
		04	00095		RET		

; Routine Size: 150 bytes, Routine Base: CODE + 1BEC

```

3557 4475 1 %sbttl 'GET_JOB_NAME - Get Name of the Current Job'
3558 4476 1 ++
3559 4477 1 Functional Description:
3560 4478 1 This routine creates a phrase with the name of the current job.
3561 4479 1
3562 4480 1 Formal Parameters:
3563 4481 1 SCB Address of the SCB
3564 4482 1 STR_DESC Desc of String to Return
3565 4483 1 RET_LEN Return length of Desc.
3566 4484 1
3567 4485 1 Implicit Inputs:
3568 4486 1 none
3569 4487 1
3570 4488 1 Implicit Outputs:
3571 4489 1 none
3572 4490 1
3573 4491 1 Returned Value:
3574 4492 1 none
3575 4493 1
3576 4494 1 Side Effects:
3577 4495 1 none
3578 4496 1 --
3579 4497 1 ROUTINE GET_JOB_NAME (
3580 4498 1 SCB : REF $BLOCK, !SCB
3581 4499 1 STR_DESC : REF VECTOR[2], !Output buffer desc
3582 4500 1 RET_LEN : REF VECTOR [,WORD] !Return length (word)
3583 4501 1 ) : NOVALUE =
3584 4502 2 BEGIN
3585 4503 2 BIND
3586 P 4504 2 SENT80_FORMAT = $DESCRIPTOR (
3587 4505 2 '!AS');
3588 P 4507 2 SFAO ( SENT80_FORMAT,
3589 P 4508 2 RET_LEN[0], !str[size] > fetched namelen
3590 P 4509 2 STR_DESC[0],
3591 P 4510 2 SCB[PSMSQ_JOB_NAME], !job name
3592 4511 2
3593 4512 2
3594 4513 2 RETURN SSS_NORMAL;
3595 4514 1 END;

```

53 41 21 01C82 P.AC0: .ASCII \!AS\
 00000003 01C85 .BLKB 3
 00000000 01C88 P.ACN: .LONG 3
 00000000 01C8C .ADDRESS P.AC0

SENT80_FORMAT= P.ACN

				0000 00000 GET_JOB_NAME:	
7E	04	AC 000000A8	8F C1 00002	.WORD	Save nothing
			08 AC DD 0000B	ADDL3	#168, SCB, -(SP)
			0C AC DD 0000E	PUSHL	STR_DESC
			E4 AF 9F 00011	PUSHL	RET_LEN
				PUSHAB	SENT80_FORMAT

: 4497
: 4511

SEPARATE
V04-001

Print Symbiont -- separation routines
GET_JOB_NAME - Get Name of the Current Job

F 6
16-Sep-1984 02:23:03 14-Sep-1984 22:32:26 VAX-11 Bliss-32 V4.0-742
[PRTSMB.SRC]SEPARATE.B32;2

Page 118
(30)

00000000G 00

04 FB 00014
04 0001B

CALLS #4, SYSSFAO
RET

; 4514

; Routine Size: 28 bytes, Routine Base: CODE + 1C90

SE
VO

```

3598 4515 1 %sbttl 'GET_EOJ - Get the Phrase End of Job'
3599 4516 1 ++
3600 4517 1 Functional Description:
3601 4518 1 This routine creates a phrase with "EOJ" or "END OF JOB".
3602 4519 1
3603 4520 1 Formal Parameters:
3604 4521 1 SCB           - Address of the SCB
3605 4522 1 STR_DESC     - Desc of String to Return
3606 4523 1 RET_LEN      - Return length of Desc.
3607 4524 1
3608 4525 1 Implicit Inputs:
3609 4526 1             none
3610 4527 1
3611 4528 1 Implicit Outputs:
3612 4529 1             none
3613 4530 1
3614 4531 1 Returned Value:
3615 4532 1             none
3616 4533 1
3617 4534 1 Side Effects:
3618 4535 1             none
3619 4536 1 --
3620 4537 1 ROUTINE GET_EOJ (
3621 4538 1     SCB          : REF SBBLOCK,          ! SCB
3622 4539 1     STR_DESC    : REF VECTOR[?],        ! Output buffer desc
3623 4540 1     RET_LEN     : REF VECTOR [,WORD]   ! Return length (word)
3624 4541 1             ) : NOVALUE =
3625 4542 2 BEGIN
3626 4543 2 BIND
3627 P 4544 2 SENT132 FORMAT = $DESCRIPTOR (
3628 4545 2             'END OF JOB').           ! -
3629 4546 2
3630 P 4547 2 SENT80 FORMAT = $DESCRIPTOR (
3631 4548 2             'EOJ');
3632 4549 2
3633 P 4550 2 $FAO ( SENT132 FORMAT,
3634 4551 2             RET_LEN[0]          ! return length
3635 4552 2             STR_DESC[0].          ! address of string
3636 4553 2         );
3637 4554 2
3638 4555 2 ! Is it short enough to allow the words "End of Job" to be printed ?
3639 4556 2
3640 4557 3 IF ((12 * .RET_LEN[0]) GTR .SCB[PSMSL_PAGE_WIDTH])
3641 4558 2 THEN
3642 P 4559 2     $FAO ( SENT80 FORMAT,
3643 4560 2             RET_LEN[0]          ! str[size] > fetched name len
3644 P 4561 2             STR_DESC[0].
3645 4562 2
3646 4563 2
3647 4564 2 RETURN SSS_NORMAL;
3648 4565 1 END;

```

```

42 4F 4A 20 46 4F 20 44 4E 45 01CAC P.ACQ: .ASCII \END OF JOB\
01CB6 0000000A 01CB8 P.ACP: .BLKB 2
01CB8 P.ACP: .LONG 10

```

SEPARATE
V04-001

Print Symbiont -- separation routines
GET_EOJ - Get the Phrase End of Job

H 6
16-Sep-1984 02:23:03 VAX-11 Bliss-32 v4.0-742
14-Sep-1984 22:32:26 [PRTSMB.SRC]SEPARATE.B32;2

Page 120
(31)

SE
VO

00000000' 01C8C .ADDRESS P.ACQ
4A 4F 45 01CC0 P.ACS: .ASCII \EOJ\
01CC3 .BLKB 1
00000003 01CC4 P.ACR: .LONG 3
00000000' 01CC8 .ADDRESS P.ACS

SENT132 FORMAT= P.ACQ
SENT80_FORMAT= P.ACR

		0004 00000 GET_EOJ: .WORD	Save R2	: 4537
		08 AC DD 00009	MOVAB SYSSFAO, R2	
		0C AC DD 0000C	PUSHL STR_DESC	4553
		DA AF 9F 0000F	PUSHL RET-LEN	
	62	03 FB 00012	PUSHAB SENT132 FORMAT	
	51	0C BC 3C 00015	CALLS #3, SYSSFAO	
	51	0C C4 00019	MOVZWL ARÉT-LEN, R1	4557
0200	50	04 AC D0 0001C	MULL2 #12, R1	
		51 D1 00020	MOVL SCB, R0	
		0C 15 00025	CMPL R1, 512(R0)	
		08 AC DD 00027	BLEQ 1\$	
		0C AC DD 0002A	PUSHL STR_DESC	4562
		C8 AF 9F 0002D	PUSHL RET-LEN	
	62	03 FB 00030	PUSHAB SENT80 FORMAT	
		04 00033 1\$:	CALLS #3, SYSSFAO	
			RET	4565

; Routine Size: 52 bytes, Routine Base: CODE + 1CCC

```

3650 4566 1 %sbttl 'GET_EOF - Get the Phrase End of File'
3651 4567 1 ++
3652 4568 1 Functional Description:
3653 4569 1 This routine creates a phrase with "EOF" or "END OF FILE".
3654 4570 1
3655 4571 1 Formal Parameters:
3656 4572 1 SCB      - Address of the SCB
3657 4573 1 STR_DESC - Desc of String to Return
3658 4574 1 RET_LEN  - Return length of Desc.
3659 4575 1
3660 4576 1 Implicit Inputs:
3661 4577 1      none
3662 4578 1
3663 4579 1 Implicit Outputs:
3664 4580 1      none
3665 4581 1
3666 4582 1 Returned Value:
3667 4583 1      none
3668 4584 1
3669 4585 1 Side Effects:
3670 4586 1      none
3671 4587 1 --
3672 4588 1 ROUTINE GET_EOF (
3673 4589 1      SCB      : REF $BBLOCK           | SCB
3674 4590 1      STR_DESC : REF VECTOR[2],        | Output buffer desc
3675 4591 1      RET_LEN  : REF VECTOR [,WORD]    | Return length (word)
3676 4592 1      )      : NOVALUE =
3677 4593 2 BEGIN
3678 4594 2 BIND
3679 P 4595 2      SENT132 FORMAT = $DESCRIPTOR (
3680 4596 2      'END OF FILE').          ! -
3681 4597 2
3682 P 4598 2      SENT80 FORMAT = $DESCRIPTOR (
3683 4599 2      'EOF');
3684 4600 2
3685 P 4601 2      SFAD ( SENT132 FORMAT,
3686 4602 2      RET_LEN[0],            ! return length
3687 4603 2      STR_DESC[0],          ! address of string
3688 4604 2      );
3689 4605 2
3690 4606 2      ! Is it short enough to allow the words "End of Job" to be printed ?
3691 4607 2
3692 4608 3      IF ((12 * .RET_LEN[0]) GTR .SCB[PSMSL_PAGE_WIDTH])
3693 4609 2      THEN
3694 P 4610 2      SFAD ( SENT80 FORMAT,
3695 4611 2      RET_LEN[0],            ! str[size] > fetched name len
3696 P 4612 2      STR_DESC[0],
3697 4613 2      );
3698 4614 2
3699 4615 2      RETURN SSS_NORMAL;
3700 4616 1      END;

```

```

45 4C 49 46 20 46 4F 20 44 4E 45 01D00 P.ACU: .ASCII \END OF FILE\
01DOB               .BLKB   1
0000000B 01DOC P.ACT: .LONG  11

```

SEPARATE
V04-001

Print Symbiant -- separation routines
GET_EOF - Get the Phrase End of File

J 6
16-Sep-1984 02:23:03 VAX-11 Blise-32 v4.0-742
14-Sep-1984 22:32:26 [PRTSMB.SRC]SEPARATE.B32;2

Page 122
(32)

SE
VO

00000000' 01D10 .ADDRESS P.ACW
46 4F 45 01D14 P.ACW: .ASCII \EOF\
01D17 .BLKB 1
00000003' 01D18 P.ACV: .LONG 3
00000000' 01D1C .ADDRESS P.ACW

SENT132 FORMAT= P.AC
SENT80_FORMAT= P.ACV

		0004 00000 GET_EOF: .WORD	Save R2	4588
		00 9E 00002	MOVAB SYSSFAO, R2	
		08 AC DD 00009	PUSHL STR_DESC	4604
		0C AC DD 0000C	PUSHL RET_LEN	
		DA AF 9F 0000F	PUSHAB SENT132 FORMAT	
	62	03 FB 00012	CALLS #3, SYSSFAO	
	51	0C BC 3C 00015	MOVZWL @RET_LEN, R1	4608
	51	0C C4 00019	MULL2 #12, R1	
0200	50 CO	04 AC DD 0001C	MOVL SCB, R0	
		51 D1 00020	CMPL R1, 512(R0)	
		0C 15 00025	BLEQ 1\$	
		08 AC DD 00027	PUSHL STR_DESC	4613
		0C AC DD 0002A	PUSHL RET_LEN	
	62	C8 AF 9F 0002D	PUSHAB SENT80 FORMAT	
		03 FB 00030	CALLS #3, SYSSFAO	
		04 00033 1\$:	RET	4616

; Routine Size: 52 bytes, Routine Base: CODE + 1D20

```

3702 4617 1 %sbttl 'GET_ACCOUNTING_INFO - Get the Accounting Information'
3703 4618 1 ++
3704 4619 1 Functional Description:
3705 4620 1 This routine returns a string containing the accounting information.
3706 4621 1
3707 4622 1 Formal Parameters:
3708 4623 1 SCB - Address of the SCB
3709 4624 1 STR_DESC - Desc of String to Return
3710 4625 1 RET_LEN - Return Length of Desc.
3711 4626 1
3712 4627 1 Implicit Inputs:
3713 4628 1 none
3714 4629 1
3715 4630 1 Implicit Outputs:
3716 4631 1 none
3717 4632 1
3718 4633 1 Returned Value:
3719 4634 1 none
3720 4635 1
3721 4636 1 Side Effects:
3722 4637 1 none
3723 4638 1 --
3724 4639 1 ROUTINE GET_ACCOUNTING_INFO (
3725 4640 1 SCB : REF $BLOCK, ! SCB
3726 4641 1 STR_DESC : REF VECTOR[,], ! Output buffer desc
3727 4642 1 RET_LEN : REF VECTOR [,WORD] ! Return length (word)
3728 4643 1 ) : NOVALUE =
3729 4644 2 BEGIN
3730 4645 2 BIND
3731 P 4646 2 SENT132 FORMAT = $DESCRIPTOR (
3732 P 4647 2 '!(AC)'.
3733 P 4648 2 '!(AS)'
3734 4649 2 );
3735 4650 2 LOCAL
3736 4651 2 IF_PRES;
3737 4652 2
3738 4653 2 IF_PRES = .SCB_SIZE_(ACCOUNTING_DATA);
3739 4654 2 IF .IF_PRES GEQ 1
3740 4655 2 THEN
3741 4656 2 IF_PRES = 1;
3742 4657 2
3743 P 4658 2 SFAD ( SENT132 FORMAT,
3744 P 4659 2 RET_LEN[0] ! return length
3745 P 4660 2 STR_DESC[0], ! address of string
3746 P 4661 2 .IF_PRES,
3747 P 4662 2 UPLIT BYTE (%ASCIC 'ACCOUNTING INFO:'),
3748 P 4663 2 .IF_PRES,
3749 P 4664 2 SCB[PSMSQ_ACCOUNTING_DATA] ! accounting data
3750 4665 2 );
3751 4666 1 END:

```

29	43	41	28	23	21	01D54	P.ACY:	.ASCII	\!(AC)\
29	53	41	28	23	21	01D5A		.ASCII	\!(AS)\
						0000000C	P.ACX:	.LONG	12
						00000000	01D60	.ADDRESS	P.ACY
						01D64			

SEPARATE
V04-001

Print Symbiont -- separation routines
GET_ACCOUNTING_INFO - Get the Accounting Inform

L 6

16-Sep-1984 02:23:03

14-Sep-1984 22:32:26

VAX-11 Bliss-32 V4.0-742
[PRTSMB.SRC]SEPARATE.B32;2

Page 124
(33)

46 4E 49 20 47 4E 49 54 4E 55 4F 43 43 41 10 01D68 P.ACZ: .ASCII <16>\ACCOUNTING INFO:\
3A 4F 01D77

SENT132_FORMAT= P.ACX

0000 00000 GET_ACCOUNTING_INFO:

50	04	AC	DD	00002	.WORD	Save nothing	: 4639
51	14	A0	3C	00006	MOVL	SCB, R0	: 4653
		03	15	0000A	MOVZWL	20(R0), IF_PRES	
51	01	DD	0000C		BLEQ	1\$: 4654
	14	A0	9F	0000F	1\$:	MOVL #1 IF_PRES	: 4656
		51	DD	00012	PUSHAB	20(R0)-	: 4665
D8	AF	9F	00014		PUSHL	IF PRES	
	08	AC	DD	00017	PUSHAB	P.ACZ	
	0C	AC	DD	00019	PUSHL	IF PRES	
C5	AF	9F	0001C		PUSHL	STR_DESC	
0000000G 00	07	FB	00022		PUSHL	RET-LEN	
	04	00029			PUSHAB	SENT132 FORMAT	
					CALLS #7, SYSSFAO		
					RET		: 4666

; Routine Size: 42 bytes, Routine Base: CODE + 1D79

```

3753 4667 1 %sbttl 'GET_QUALIFIERS - Get Switches/Qualifiers associated with PRINTING'
3754 4668 1 ++
3755 4669 1 Functional Description:
3756 4670 1 This routine returns a string containing the all relevant print
3757 4671 1 qualifier information.
3758 4672 1
3759 4673 1 Formal Parameters:
3760 4674 1 SCB - Address of the SCB
3761 4675 1 STR_DESC - Desc of String to Return
3762 4676 1 RET_LEN - Return Length of Desc.
3763 4677 1
3764 4678 1 Implicit Inputs:
3765 4679 1 none
3766 4680 1
3767 4681 1 Implicit Outputs:
3768 4682 1 none
3769 4683 1
3770 4684 1 Returned Value:
3771 4685 1 none
3772 4686 1
3773 4687 1 Side Effects:
3774 4688 1 none
3775 4689 1 --
3776 4690 1 ROUTINE GET_QUALIFIERS (
3777 4691 1 SCB : REF $BLOCK, : SCB
3778 4692 1 STR_DESC : REF VECTOR[2], : Output buffer desc
3779 4693 1 RET_LEN : REF VECTOR [,WORD] : Return length (word)
3780 4694 1 ) : NOVALUE =
3781 4695 2 BEGIN
3782 4696 2 BIND
3783 4697 2 ! all the formats start here
3784 4698 2
3785 P 4699 2 DATE_FORMAT = $DESCRIPTOR (
3786 4700 2 '!17%D'),
3787 4701 2
3788 P 4702 2 AFTER_TIME_FORMAT = $DESCRIPTOR (
3789 4703 2 '-/AFTER=!17%D'), ! - after_time print
3790 4704 2
3791 P 4705 2 BURST_FORMAT = $DESCRIPTOR (
3792 4706 2 '/BURST'), ! - burst
3793 4707 2
3794 P 4708 2 CHARACTERISTICS_FORMAT = $DESCRIPTOR (
3795 4709 2 '/CHARACTERISTICS=!AS'), ! - characteristics
3796 4710 2
3797 P 4711 2 FILE_COPIES_FORMAT = $DESCRIPTOR (
3798 4712 2 '/COPIES=!UL(!UL of !UL)'), ! - copies/iteration
3799 4713 2
3800 P 4714 2 FEED_FORMAT = $DESCRIPTOR (
3801 4715 2 '/FEED'), ! - feed
3802 4716 2
3803 P 4717 2 FLAG_FORMAT = $DESCRIPTOR (
3804 4718 2 '/FLAG'), ! - flag
3805 4719 2
3806 P 4720 2 FORM_FORMAT = $DESCRIPTOR (
3807 4721 2 '/FORM=!AS'), ! - form
3808 4722 2
3809 P 4723 2 HEADER_FORMAT = $DESCRIPTOR (

```

SEPARATE
 V04-001 Print Symbiont -- separation routines
 GET_QUALIFIERS - Get Switches/Qualifiers associ 16-Sep-1984 02:23:03
 14-Sep-1984 22:32:26 VAX-11 Bliss-32 v4.0-742
 [PRTSMB.SRC]SEPARATE.B32;2 Page 126 (34)

```

 3810      4724 2     ' /HEADER'           ),          ! - header
 3811      4725 2
 3812      P 4726 2     JOB_COUNT FORMAT = $DESCRIPTOR (
 3813          ' /JOB_COUNT=!UL(!UL of !UL)'),      ! - job count /iteration
 3814      4727 2
 3815      P 4729 2     LENGTH FORMAT = $DESCRIPTOR(
 3816          ' !LENGTH=!UL'),
 3817      4730 2
 3818      P 4732 2     LIBRARY FORMAT = $DESCRIPTOR (
 3819          ' /LIBRARY=!AS' ),
 3820      4733 2
 3821      P 4734 2     MARGIN FORMAT = $DESCRIPTOR(
 3822          ' !MARGIN=()'),
 3823      4735 2
 3824      P 4737 2     TOP FORMAT = $DESCRIPTOR(
 3825          ' !TOP=!UL'),
 3826      4738 2
 3827      P 4741 2     BOTTOM FORMAT = $DESCRIPTOR(
 3828          ' !BOTTOM=!UL'),
 3829      4742 2
 3830      P 4743 2     LEFT FORMAT = $DESCRIPTOR(
 3831          ' !LEFT=!UL'),
 3832      4744 2
 3833      P 4746 2     RIGHT FORMAT = $DESCRIPTOR(
 3834          ' !RIGHT=!UL'),
 3835      4747 2
 3836      P 4750 2     NOFEED FORMAT = $DESCRIPTOR (
 3837          ' !NOFEED' ),
 3838      4751 2
 3839      P 4753 2     SETUP_PAGE FORMAT = $DESCRIPTOR(
 3840          ' /PAGE_SETUP=(!AS)' ),
 3841      4754 2
 3842      P 4756 2     PAGES FORMAT = $DESCRIPTOR (
 3843          ' /PAGES=(!UL,!UL)' ),
 3844      4757 2
 3845      P 4759 2     PARAMETER FORMAT = $DESCRIPTOR (
 3846          ' /PARAMETERS=(',
 3847          ' ""!AS", ""!AS", ""!AS", ""!AS", ""!AS", ',
 3848          ' ""!AS", ""!AS", ""!AS")'),
 3849      4760 2
 3850      P 4764 2     PASSALL FORMAT = $DESCRIPTOR (
 3851          ' /PASSALL' ),
 3852      4765 2
 3853      P 4766 2     PUNCTUATION FORMAT = $DESCRIPTOR(
 3854          ' !AC'),
 3855      4768 2
 3856      P 4770 2     SETUP_FILE FORMAT = $DESCRIPTOR(
 3857          ' /SETUP_FILE=(!AS)' ),
 3858      4771 2
 3859      P 4773 2     SETUP_FORM FORMAT = $DESCRIPTOR(
 3860          ' /SETUP_FORM=(!AS)' ),
 3861      4774 2
 3862      P 4776 2     SHEET_FORMAT = $DESCRIPTOR(
 3863          ' /SHEET_FEED' ),
 3864      4777 2
 3865      P 4779 2     SPACE_FORMAT = $DESCRIPTOR(
 3866          ' /SPACE' ),
 3867      4780 2
 3868
 3869
 3870
 3871
 3872
 3873
 3874
 3875
 3876
 3877
 3878
 3879
 3880
 3881
 3882
 3883
 3884
 3885
 3886
 3887
 3888
 3889
 3890
 3891
 3892
 3893
 3894
 3895
 3896
 3897
 3898
 3899
 3900
 3901
 3902
 3903
 3904
 3905
 3906
 3907
 3908
 3909
 3910
 3911
 3912
 3913
 3914
 3915
 3916
 3917
 3918
 3919
 3920
 3921
 3922
 3923
 3924
 3925
 3926
 3927
 3928
 3929
 3930
 3931
 3932
 3933
 3934
 3935
 3936
 3937
 3938
 3939
 3940
 3941
 3942
 3943
 3944
 3945
 3946
 3947
 3948
 3949
 3950
 3951
 3952
 3953
 3954
 3955
 3956
 3957
 3958
 3959
 3960
 3961
 3962
 3963
 3964
 3965
 3966
 3967
 3968
 3969
 3970
 3971
 3972
 3973
 3974
 3975
 3976
 3977
 3978
 3979
 3980
 3981
 3982
 3983
 3984
 3985
 3986
 3987
 3988
 3989
 3990
 3991
 3992
 3993
 3994
 3995
 3996
 3997
 3998
 3999
 4000
 4001
 4002
 4003
 4004
 4005
 4006
 4007
 4008
 4009
 4010
 4011
 4012
 4013
 4014
 4015
 4016
 4017
 4018
 4019
 4020
 4021
 4022
 4023
 4024
 4025
 4026
 4027
 4028
 4029
 4030
 4031
 4032
 4033
 4034
 4035
 4036
 4037
 4038
 4039
 4040
 4041
 4042
 4043
 4044
 4045
 4046
 4047
 4048
 4049
 4050
 4051
 4052
 4053
 4054
 4055
 4056
 4057
 4058
 4059
 4060
 4061
 4062
 4063
 4064
 4065
 4066
 4067
 4068
 4069
 4070
 4071
 4072
 4073
 4074
 4075
 4076
 4077
 4078
 4079
 4080
 4081
 4082
 4083
 4084
 4085
 4086
 4087
 4088
 4089
 4090
 4091
 4092
 4093
 4094
 4095
 4096
 4097
 4098
 4099
 4100
 4101
 4102
 4103
 4104
 4105
 4106
 4107
 4108
 4109
 4110
 4111
 4112
 4113
 4114
 4115
 4116
 4117
 4118
 4119
 4120
 4121
 4122
 4123
 4124
 4125
 4126
 4127
 4128
 4129
 4130
 4131
 4132
 4133
 4134
 4135
 4136
 4137
 4138
 4139
 4140
 4141
 4142
 4143
 4144
 4145
 4146
 4147
 4148
 4149
 4150
 4151
 4152
 4153
 4154
 4155
 4156
 4157
 4158
 4159
 4160
 4161
 4162
 4163
 4164
 4165
 4166
 4167
 4168
 4169
 4170
 4171
 4172
 4173
 4174
 4175
 4176
 4177
 4178
 4179
 4180
 4181
 4182
 4183
 4184
 4185
 4186
 4187
 4188
 4189
 4190
 4191
 4192
 4193
 4194
 4195
 4196
 4197
 4198
 4199
 4200
 4201
 4202
 4203
 4204
 4205
 4206
 4207
 4208
 4209
 4210
 4211
 4212
 4213
 4214
 4215
 4216
 4217
 4218
 4219
 4220
 4221
 4222
 4223
 4224
 4225
 4226
 4227
 4228
 4229
 4230
 4231
 4232
 4233
 4234
 4235
 4236
 4237
 4238
 4239
 4240
 4241
 4242
 4243
 4244
 4245
 4246
 4247
 4248
 4249
 4250
 4251
 4252
 4253
 4254
 4255
 4256
 4257
 4258
 4259
 4260
 4261
 4262
 4263
 4264
 4265
 4266
 4267
 4268
 4269
 4270
 4271
 4272
 4273
 4274
 4275
 4276
 4277
 4278
 4279
 4280
 4281
 4282
 4283
 4284
 4285
 4286
 4287
 4288
 4289
 4290
 4291
 4292
 4293
 4294
 4295
 4296
 4297
 4298
 4299
 4300
 4301
 4302
 4303
 4304
 4305
 4306
 4307
 4308
 4309
 4310
 4311
 4312
 4313
 4314
 4315
 4316
 4317
 4318
 4319
 4320
 4321
 4322
 4323
 4324
 4325
 4326
 4327
 4328
 4329
 4330
 4331
 4332
 4333
 4334
 4335
 4336
 4337
 4338
 4339
 4340
 4341
 4342
 4343
 4344
 4345
 4346
 4347
 4348
 4349
 4350
 4351
 4352
 4353
 4354
 4355
 4356
 4357
 4358
 4359
 4360
 4361
 4362
 4363
 4364
 4365
 4366
 4367
 4368
 4369
 4370
 4371
 4372
 4373
 4374
 4375
 4376
 4377
 4378
 4379
 4380
 4381
 4382
 4383
 4384
 4385
 4386
 4387
 4388
 4389
 4390
 4391
 4392
 4393
 4394
 4395
 4396
 4397
 4398
 4399
 4400
 4401
 4402
 4403
 4404
 4405
 4406
 4407
 4408
 4409
 4410
 4411
 4412
 4413
 4414
 4415
 4416
 4417
 4418
 4419
 4420
 4421
 4422
 4423
 4424
 4425
 4426
 4427
 4428
 4429
 4430
 4431
 4432
 4433
 4434
 4435
 4436
 4437
 4438
 4439
 4440
 4441
 4442
 4443
 4444
 4445
 4446
 4447
 4448
 4449
 4450
 4451
 4452
 4453
 4454
 4455
 4456
 4457
 4458
 4459
 4460
 4461
 4462
 4463
 4464
 4465
 4466
 4467
 4468
 4469
 4470
 4471
 4472
 4473
 4474
 4475
 4476
 4477
 4478
 4479
 4480
 4481
 4482
 4483
 4484
 4485
 4486
 4487
 4488
 4489
 4490
 4491
 4492
 4493
 4494
 4495
 4496
 4497
 4498
 4499
 4500
 4501
 4502
 4503
 4504
 4505
 4506
 4507
 4508
 4509
 4510
 4511
 4512
 4513
 4514
 4515
 4516
 4517
 4518
 4519
 4520
 4521
 4522
 4523
 4524
 4525
 4526
 4527
 4528
 4529
 4530
 4531
 4532
 4533
 4534
 4535
 4536
 4537
 4538
 4539
 4540
 4541
 4542
 4543
 4544
 4545
 4546
 4547
 4548
 4549
 4550
 4551
 4552
 4553
 4554
 4555
 4556
 4557
 4558
 4559
 4560
 4561
 4562
 4563
 4564
 4565
 4566
 4567
 4568
 4569
 4570
 4571
 4572
 4573
 4574
 4575
 4576
 4577
 4578
 4579
 4580
 4581
 4582
 4583
 4584
 4585
 4586
 4587
 4588
 4589
 4590
 4591
 4592
 4593
 4594
 4595
 4596
 4597
 4598
 4599
 4600
 4601
 4602
 4603
 4604
 4605
 4606
 4607
 4608
 4609
 4610
 4611
 4612
 4613
 4614
 4615
 4616
 4617
 4618
 4619
 4620
 4621
 4622
 4623
 4624
 4625
 4626
 4627
 4628
 4629
 4630
 4631
 4632
 4633
 4634
 4635
 4636
 4637
 4638
 4639
 4640
 4641
 4642
 4643
 4644
 4645
 4646
 4647
 4648
 4649
 4650
 4651
 4652
 4653
 4654
 4655
 4656
 4657
 4658
 4659
 4660
 4661
 4662
 4663
 4664
 4665
 4666
 4667
 4668
 4669
 4670
 4671
 4672
 4673
 4674
 4675
 4676
 4677
 4678
 4679
 4680
 4681
 4682
 4683
 4684
 4685
 4686
 4687
 4688
 4689
 4690
 4691
 4692
 4693
 4694
 4695
 4696
 4697
 4698
 4699
 4700
 4701
 4702
 4703
 4704
 4705
 4706
 4707
 4708
 4709
 4710
 4711
 4712
 4713
 4714
 4715
 4716
 4717
 4718
 4719
 4720
 4721
 4722
 4723
 4724
 4725
 4726
 4727
 4728
 4729
 4730
 4731
 4732
 4733
 4734
 4735
 4736
 4737
 4738
 4739
 4740
 4741
 4742
 4743
 4744
 4745
 4746
 4747
 4748
 4749
 4750
 4751
 4752
 4753
 4754
 4755
 4756
 4757
 4758
 4759
 4760
 4761
 4762
 4763
 4764
 4765
 4766
 4767
 4768
 4769
 4770
 4771
 4772
 4773
 4774
 4775
 4776
 4777
 4778
 4779
 4780
 4781
 4782
 4783
 4784
 4785
 4786
 4787
 4788
 4789
 4790
 4791
 4792
 4793
 4794
 4795
 4796
 4797
 4798
 4799
 4800
 4801
 4802
 4803
 4804
 4805
 4806
 4807
 4808
 4809
 4810
 4811
 4812
 4813
 4814
 4815
 4816
 4817
 4818
 4819
 4820
 4821
 4822
 4823
 4824
 4825
 4826
 4827
 4828
 4829
 4830
 4831
 4832
 4833
 4834
 4835
 4836
 4837
 4838
 4839
 4840
 4841
 4842
 4843
 4844
 4845
 4846
 4847
 4848
 4849
 4850
 4851
 4852
 4853
 4854
 4855
 4856
 4857
 4858
 4859
 4860
 4861
 4862
 4863
 4864
 4865
 4866
 4867
 4868
 4869
 4870
 4871
 4872
 4873
 4874
 4875
 4876
 4877
 4878
 4879
 4880
 4881
 4882
 4883
 4884
 4885
 4886
 4887
 4888
 4889
 4890
 4891
 4892
 4893
 4894
 4895
 4896
 4897
 4898
 4899
 4900
 4901
 4902
 4903
 4904
 4905
 4906
 4907
 4908
 4909
 4910
 4911
 4912
 4913
 4914
 4915
 4916
 4917
 4918
 4919
 4920
 4921
 4922
 4923
 4924
 4925
 4926
 4927
 4928
 4929
 4930
 4931
 4932
 4933
 4934
 4935
 4936
 4937
 4938
 4939
 4940
 4941
 4942
 4943
 4944
 4945
 4946
 4947
 4948
 4949
 4950
 4951
 4952
 4953
 4954
 4955
 4956
 4957
 4958
 4959
 4960
 4961
 4962
 4963
 4964
 4965
 4966
 4967
 4968
 4969
 4970
 4971
 4972
 4973
 4974
 4975
 4976
 4977
 4978
 4979
 4980
 4981
 4982
 4983
 4984
 4985
 4986
 4987
 4988
 4989
 4990
 4991
 4992
 4993
 4994
 4995
 4996
 4997
 4998
 4999
 5000
 5001
 5002
 5003
 5004
 5005
 5006
 5007
 5008
 5009
 5010
 5011
 5012
 5013
 5014
 5015
 5016
 5017
 5018
 5019
 5020
 5021
 5022
 5023
 5024
 5025
 5026
 5027
 5028
 5029
 5030
 5031
 5032
 5033
 5034
 5035
 5036
 5037
 5038
 5039
 5040
 5041
 5042
 5043
 5044
 5045
 5046
 5047
 5048
 5049
 5050
 5051
 5052
 5053
 5054
 5055
 5056
 5057
 5058
 5059
 5060
 5061
 5062
 5063
 5064
 5065
 5066
 5067
 5068
 5069
 5070
 5071
 5072
 5073
 5074
 5075
 5076
 5077
 5078
 5079
 5080
 5081
 5082
 5083
 5084
 5085
 5086
 5087
 5088
 5089
 5090
 5091
 5092
 5093
 5094
 5095
 5096
 5097
 5098
 5099
 5100
 5101
 5102
 5103
 5104
 5105
 5106
 5107
 5108
 5109
 5110
 5111
 5112
 5113
 5114
 5115
 5116
 5117
 5118
 5119
 5120
 5121
 5122
 5123
 5124
 5125
 5126
 5127
 5128
 5129
 5130
 5131
 5132
 5133
 5134
 5135
 5136
 5137
 5138
 5139
 5140
 5141
 5142
 5143
 5144
 5145
 5146
 5147
 5148
 5149
 5150
 5151
 5152
 5153
 5154
 5155
 5156
 5157
 5158
 5159
 5160
 5161
 5162
 5163
 5164
 5165
 5166
 5167
 5168
 5169
 5170
 5171
 5172
 5173
 5174
 5175
 5176
 5177
 5178
 5179
 5180
 5181
 5182
 5183
 5184
 5185
 5186
 5187
 5188
 5189
 5190
 5191
 5192
 5193
 5194
 5195
 5196
 5197
 5198
 5199
 5200
 5201
 5202
 5203
 5204
 5205
 5206
 5207
 5208
 5209
 5210
 5211
 5212
 5213
 5214
 5215
 5216
 5217
 5218
 5219
 5220
 5221
 5222
 5223
 5224
 5225
 5226
 5227
 5228
 5229
 5230
 5231
 5232
 5233
 5234
 5235
 5236
 5237
 5238
 5239
 5240
 5241
 5242
 5243
 5244
 5245
 5246
 5247
 5248
 5249
 5250
 5251
 5252
 5253
 5254
 5255
 5256
 5257
 5258
 5259
 5260
 5261
 5262
 5263
 5264
 5265
 5266
 5267
 5268
 5269
 5270
 5271
 5272
 5273
 5274
 5275
 5276
 5277
 5278
 5279
 5280
 5281
 5282

```

```

: 3867      4781 2
: 3868      P 4782 2      TRAILER FORMAT = $DESCRIPTOR(
: 3869          4783 2      'TRAILER' ),
: 3870          4784 2      ! - trailer
: 3871      P 4785 2      TRUNCATE FORMAT = $DESCRIPTOR(
: 3872          4786 2      'TRUNCATE' ),
: 3873          4787 2
: 3874      P 4788 2      WIDTH_FORMAT = $DESCRIPTOR(
: 3875          4789 2      '/WIDTH=!UL'),
: 3876          4790 2
: 3877      P 4791 2      WRAP FORMAT = $DESCRIPTOR(
: 3878          4792 2      '/WRAP' );
: 3879          4793 2
: 3880          4794 2      LITERAL
: 3881          4795 2      K_MAX_BUFFER_SIZE = 512;
: 3882          4796 2
: 3883          4797 2      LOCAL
: 3884          4798 2      PUNC_FLAG : INITIAL (0),
: 3885          4799 2      TEMP_LEN
: 3886          4800 2      AFT_DATE_PTR: VECTOR[2],
: 3887          4801 2      TEMP_PTR : VECTOR[2],
: 3888          4802 2      AFT_BUFF : VECTOR[17,byte],
: 3889          4803 2      TEMP_BUFF : VECTOR[17,byte],
: 3890          4804 2      IF PRES
: 3891          4805 2      CURRENT_LEN : INITIAL (0),
: 3892          4806 2      STRING_PTR : VECTOR [2];           ! Pointer to current string
: 3893          4807 2
: 3894          4808 2      ! Allocate the buffer for "GET_xxx" Routines
: 3895          4809 2
: 3896          4810 2      STRING_PTR[SIZE] = K_MAX_BUFFER_SIZE;
: 3897          4811 2      STRING_PTR[ADDR] = .STR_DESC[ADDR];   ! init address
: 3898          4812 2
: 3899          4813 2      RET_LEN[0] = 0;
: 3900          4814 2
: 3901          4815 2      !$FAO ( BEGIN_FORMAT,
: 3902          4816 2      CURRENT_LEN,
: 3903          4817 2      STRING_PTR[0],                      ! return length
: 3904          4818 2      ! );                                ! address of string
: 3905          4819 2
: 3906          4820 2      !RET_LEN[0] = .RET_LEN[0] + .CURRENT_LEN;
: 3907          4821 2      !STRING_PTR[ADDR] = .STRING_PTR[ADDR] + .CURRENT_LEN;
: 3908          4822 2      !STRING_PTR[SIZE] = K_MAX_BUFFER_SIZE - .RET_LEN[0];
: 3909          4823 2
: 3910          4824 2
: 3911          4825 2      ! Increment pointer only if not equal to time queued
: 3912          4826 2
: 3913          4827 2      AFT_DATE_PTR[SIZE] = XALLOCATION(AFT_BUFF);
: 3914          4828 2      AFT_DATE_PTR[ADDR] = AFT_BUFF;
: 3915          4829 2
: 3916          P 4830 2      $FAO ( DATE_FORMAT,
: 3917          P 4831 2      TEMP_LEN,
: 3918          P 4832 2      AFT_DATE_PTR[0],
: 3919          4833 2      SCB[PSMSQ_AFTER_TIME]);
: 3920          4834 2
: 3921          4835 2      TEMP_PTR[SIZE] = XALLOCATION(TEMP_BUFF);
: 3922          4836 2      TEMP_PTR[ADDR] = TEMP_BUFF;
: 3923          4837 2

```

```
: 3924 P 4838 2 $FAO ( DATE_FORMAT,
: 3925 P 4839 2 TEMP_LEN
: 3926 P 4840 2 TEMP_PTR[0]
: 3927 P 4841 2 SCB[PSMSQ_TIME_QUEUE];
: 3928 P 4842 2
: 3929 P 4843 2 IF CHSNEQ( .TEMP_LEN, .TEMP_PTR[ADDR], .TEMP_LEN, .AFT_DATE_PTR[ADDR])
: 3930 P 4844 2 THEN BEGIN
: 3931 P 4845 2
: 3932 P 4846 2
: 3933 P 4847 2 $FAO (
: 3934 P 4848 2 AFTER_TIME_FORMAT,
: 3935 P 4849 2 CURRENT_LEN
: 3936 P 4850 2 STRING_PTR[0],
: 3937 P 4851 2 SCB[PSMSQ_AFTERTIME]); : return length
: 3938 P 4852 2 : address of string
: 3939 P 4853 2 : after_time
: 3940 P 4854 2 RET_LEN[0] = .RET_LEN[0] + .CURRENT_LEN;
: 3941 P 4855 2 STRING_PTR[ADDR] = .STRING_PTR[ADDR] + .CURRENT_LEN;
: 3942 P 4856 2 STRING_PTR[SIZE] = K_MAX_BUFFER_SIZE - .RET_LEN[0];
: 3943 P 4857 2 END;
: 3944 P 4858 2 IF .SEPARATE_FLAG_ (FILE_BURST)
: 3945 P 4859 2 THEN BEGIN
: 3946 P 4860 2
: 3947 P 4861 2 $FAO ( BURST_FORMAT,
: 3948 P 4862 2 CURRENT_LEN
: 3949 P 4863 2 STRING_PTR[0]
: 3950 P 4864 2 );
: 3951 P 4865 2
: 3952 P 4866 2 RET_LEN[0] = .RET_LEN[0] + .CURRENT_LEN;
: 3953 P 4867 2 STRING_PTR[ADDR] = .STRING_PTR[ADDR] + .CURRENT_LEN;
: 3954 P 4868 2 STRING_PTR[SIZE] = K_MAX_BUFFER_SIZE - .RET_LEN[0];
: 3955 P 4869 2 END;
: 3956 P 4870 2
: 3957 P 4871 2 IF (.SCB_SIZE_ (CHARACTERISTICS) EQL 0)
: 3958 P 4872 2 THEN BEGIN
: 3959 P 4873 2
: 3960 P 4874 2 $FAO ( CHARACTERISTICS_FORMAT,
: 3961 P 4875 2 CURRENT_LEN
: 3962 P 4876 2 STRING_PTR[0],
: 3963 P 4877 2 SCB[PSMSQ_CHARACTERISTICS] : return length
: 3964 P 4878 2 : address of string
: 3965 P 4879 2 : /CHARACTERISTICS
: 3966 P 4880 2
: 3967 P 4881 2 RET_LEN[0] = .RET_LEN[0] + .CURRENT_LEN;
: 3968 P 4882 2 STRING_PTR[ADDR] = .STRING_PTR[ADDR] + .CURRENT_LEN;
: 3969 P 4883 2 STRING_PTR[SIZE] = K_MAX_BUFFER_SIZE - .RET_LEN[0];
: 3970 P 4884 2 END;
: 3971 P 4885 2 ! Always print something about form feed... /FEED or /NOFEED
: 3972 P 4886 2
: 3973 P 4887 2 IF .$BBLOCK[SCB[PSMSL_PRINT_CONTROL], SMBMSG$V_PAGINATE]
: 3974 P 4888 2 THEN BEGIN
: 3975 P 4889 2
: 3976 P 4890 2 $FAO ( FEED_FORMAT,
: 3977 P 4891 2 CURRENT_LEN
: 3978 P 4892 2 STRING_PTR[0]
: 3979 P 4893 2 );
: 3980 P 4894 2
```

3981 4895 3 RET_LEN[0] = .RET_LEN[0] + .CURRENT_LEN;
3982 4896 3 STRING_PTR[ADDR] = .STRING_PTR[ADDR] + .CURRENT_LEN;
3983 4897 3 STRING_PTR[SIZE] = K_MAX_BUFFER_SIZE - .RET_LEN[0];
3984 4898 2 END;
3985 4899 2
3986 4900 2 IF .SCB[PSMSL_FILE_COPIES] GTR 1
3987 4901 2 THEN
3988 4902 3 BEGIN
3989 P 4903 3 \$FAO (FILE_COPIES_FORMAT,
3990 P 4904 3 CURRENT_LEN,
3991 P 4905 3 STRING_PTR[0],
3992 P 4906 3 .SCB[PSMSL_FILE_COPIES],
3993 P 4907 3 .SCB[PSMSL_FILE_COUNT],
3994 P 4908 3 .SCB[PSMSL_FILE_COPIES]
3995 4909 3);
3996 4910 3
3997 4911 3 RET_LEN[0] = .RET_LEN[0] + .CURRENT_LEN;
3998 4912 3 STRING_PTR[ADDR] = .STRING_PTR[ADDR] + .CURRENT_LEN;
3999 4913 3 STRING_PTR[SIZE] = K_MAX_BUFFER_SIZE - .RET_LEN[0];
4000 4914 2 END;
4001 4915 2
4002 4916 2 IF .SEPARATE_FLAG_ (FILE_FLAG)
4003 4917 2 THEN
4004 4918 3 BEGIN
4005 P 4919 3 \$FAO (FLAG_FORMAT,
4006 P 4920 3 CURRENT_LEN,
4007 P 4921 3 STRING_PTR[0]
4008 4922 3);
4009 4923 3
4010 4924 3 RET_LEN[0] = .RET_LEN[0] + .CURRENT_LEN;
4011 4925 3 STRING_PTR[ADDR] = .STRING_PTR[ADDR] + .CURRENT_LEN;
4012 4926 3 STRING_PTR[SIZE] = K_MAX_BUFFER_SIZE - .RET_LEN[0];
4013 4927 2 END;
4014 4928 2
4015 P 4929 2 \$FAO (FORM_FORMAT,
4016 P 4930 2 CURRENT_LEN,
4017 P 4931 2 STRING_PTR[0],
4018 P 4932 2 .SCB[PSMSQ_FORM_NAME]);
4019 4933 2
4020 4934 2 RET_LEN[0] = .RET_LEN[0] + .CURRENT_LEN;
4021 4935 2 STRING_PTR[ADDR] = .STRING_PTR[ADDR] + .CURRENT_LEN;
4022 4936 2 STRING_PTR[SIZE] = K_MAX_BUFFER_SIZE - .RET_LEN[0];
4023 4937 2
4024 4938 2 IF .RET_LEN[0] GTR K_MAX_BUFFER_SIZE
4025 4939 2 THEN
4026 4940 3 BEGIN
4027 4941 3 RET_LEN[0] = 512;
4028 4942 3 RETURN;
4029 4943 2 END;
4030 4944 2
4031 4945 2 IF .SBBLOCK[SCB[PSMSL_PRINT_CONTROL], SMBMSG\$V_PAGE_HEADER]
4032 4946 2 THEN
4033 4947 3 BEGIN
4034 P 4948 3 \$FAO (HEADER_FORMAT,
4035 P 4949 3 CURRENT_LEN,
4036 P 4950 3 STRING_PTR[0]
4037 4951 3);

```

4038      4952 3
4039      4953 3      RET_LEN[0] = .RET_LEN[0] + .CURRENT_LEN;
4040      4954 3      STRING_PTR[ADDR] = .STRING_PTR[ADDR] + .CURRENT_LEN;
4041      4955 3      STRING_PTR[SIZE] = K_MAX_BUFFER_SIZE - .RET_LEN[0];
4042      4956 2      END;
4043      4957 2
4044      4958 2      IF .RET_LEN[0] GTR K_MAX_BUFFER_SIZE
4045      4959 2      THEN
4046          4960 3      BEGIN
4047              4961 3          RET_LEN[0] = 512;
4048              4962 3          RETURN;
4049          4963 3      END;
4050      4964 2
4051      4965 2      IF .SCB[PSMSL_JOB_COPIES] GTR 1
4052      4966 2      THEN
4053          4967 3      BEGIN
4054              P 4968 3          $FAO ( JOB_COUNT FORMAT,
4055                  CURRENT_LEN,
4056                  STRING_PTR[0],                                ! return length
4057                  .SCB[PSMSL_JOB_COPIES],                      address of string
4058                  .SCB[PSMSL_JOB_COUNT],                         /JOB_COUNT
4059                  .SCB[PSMSL_JOB_COPIES]                        iteration
4060          4973 3      );
4061      4974 3
4062      4975 3      RET_LEN[0] = .RET_LEN[0] + .CURRENT_LEN;
4063      4976 3      STRING_PTR[ADDR] = .STRING_PTR[ADDR] + .CURRENT_LEN;
4064      4977 3      STRING_PTR[SIZE] = K_MAX_BUFFER_SIZE - .RET_LEN[0];
4065      4979 2      END;
4066      4980 2
4067      4981 2      ! Here is my internal call to PSMSREAD_ITEM_DX to insure that the user
4068      4982 2      can copy information successfully using this routine
4069      4983 2
4070          4984 3      BEGIN
4071              4985 3          LOCAL LEN: VECTOR[2];
4072              4986 3
4073          4987 3          INIT DYN_DESC_(LEN);
4074          4988 3          PSMSREAD_ITEM_DX (.SCB, &ref(SMBMSG$K_FORM_LENGTH),
4075                  LEN[0]);
4076          4990 3
4077          P 4991 3          $FAO ( LENGTH FORMAT,
4078              CURRENT_LEN,                                ! return length
4079              STRING_PTR[0],                                address of string
4080              ..LEN[ADDR]                                ! Length pointed to by len[addr]
4081          4995 3
4082          4996 3
4083          4997 3      RET_LEN[0] = .RET_LEN[0] + .CURRENT_LEN;
4084          4998 3      STRING_PTR[ADDR] = .STRING_PTR[ADDR] + .CURRENT_LEN;
4085          4999 3      STRING_PTR[SIZE] = K_MAX_BUFFER_SIZE - .RET_LEN[0];
4086          5000 2      END;
4087          5001 2
4088          5002 2      !*** ALWAYS PRINT THE LIBRARY !**!
4089          5003 2
4090          P 5004 2          $FAO ( LIBRARY FORMAT,
4091              CURRENT_LEN,                                ! return length
4092              STRING_PTR[0],                                address of string
4093              SCB[PSMSQ_LIBRARY_SPECIFICATION]           /LIBRARY
4094          5008 2      );

```

```
: 4095      5009 2
: 4096      5010 2 RET_LEN[0] = .RET_LEN[0] + .CURRENT_LEN;
: 4097      5011 2 STRING_PTR[ADDR] = .STRING_PTR[ADDR] + .CURRENT_LEN;
: 4098      5012 2 STRING_PTR[SIZE] = K_MAX_BUFFER_SIZE - .RET_LEN[0];
: 4099
: 4100      5013 2
: 4101      5014 2 IF .RET_LEN[0] GTR K_MAX_BUFFER_SIZE
: 4102      5015 2 THEN
: 4103          BEGIN
: 4104              RET_LEN[0] = 512;
: 4105          RETURN;
: 4106          END;
: 4107      5016 2
: 4108      5017 2
: 4109      5018 2
: 4110      5019 2
: 4111      5020 2
: 4112      5021 2 IF (.SCB[PSMSL_TOP_MARGIN] NEQ 0) OR
: 4113          (.SCB[PSMSL_BOTTOM_MARGIN] NEQ 0) OR
: 4114          (.SCB[PSMSL_LEFT_MARGIN] NEQ 0) OR
: 4115          (.SCB[PSMSL_RIGHT_MARGIN] NEQ 0)
: 4116      5022 2
: 4117      5023 2
: 4118      5024 2
: 4119      5025 2
: 4120      5026 2 THEN
: 4121          BEGIN
: 4122              $FAO ( MARGIN_FORMAT,
: 4123                  CURRENT_LEN,
: 4124                  STRING_PTR[0]
: 4125                  );
: 4126          RET_LEN[0] = .RET_LEN[0] + .CURRENT_LEN;
: 4127          STRING_PTR[ADDR] = .STRING_PTR[ADDR] + .CURRENT_LEN;
: 4128          STRING_PTR[SIZE] = K_MAX_BUFFER_SIZE - .RET_LEN[0];
: 4129      5027 2
: 4130      5028 2
: 4131      5029 2
: 4132      5030 2
: 4133      5031 2
: 4134      5032 2
: 4135      5033 2
: 4136      5034 2
: 4137      5035 2
: 4138      5036 2
: 4139      5037 2
: 4140      5038 2
: 4141      5039 2
: 4142      5040 2 IF (.SCB[PSMSL_TOP_MARGIN] NEQ 0)
: 4143      5041 2 THEN
: 4144          BEGIN
: 4145              $FAO ( TOP_FORMAT,
: 4146                  CURRENT_LEN,
: 4147                  STRING_PTR[0],
: 4148                  .SCB[PSMSL_TOP_MARGIN]
: 4149                  );
: 4150          RET_LEN[0] = .RET_LEN[0] + .CURRENT_LEN;
: 4151          STRING_PTR[ADDR] = .STRING_PTR[ADDR] + .CURRENT_LEN;
: 4152          STRING_PTR[SIZE] = K_MAX_BUFFER_SIZE - .RET_LEN[0];
: 4153      5042 2
: 4154      5043 2
: 4155      5044 2
: 4156      5045 2
: 4157      5046 2 PUNC_FLAG = 1;
: 4158      5047 2
: 4159      5048 2
: 4160      5049 2
: 4161      5050 2
: 4162      5051 2
: 4163      5052 2
: 4164      5053 2
: 4165      5054 2
: 4166      5055 2
: 4167      5056 2
: 4168      5057 2
: 4169      5058 2
: 4170      5059 2
: 4171      5060 2
: 4172      5061 2
: 4173      5062 2
: 4174      5063 2
: 4175      5064 2
: 4176      5065 2
: 4177          IF (.SCB[PSMSL_BOTTOM_MARGIN] NEQ 0) AND
: 4178              .PUNC_FLAG
: 4179          THEN
: 4180              BEGIN
: 4181                  $FAO(
: 4182                      PUNCTUATION_FORMAT,
: 4183                      CURRENT_LEN,
: 4184                      STRING_PTR[0],
: 4185                      UPLIT BYTE (%ASCIC ',')
: 4186                      );
: 4187          RET_LEN[0] = .RET_LEN[0] + .CURRENT_LEN;
: 4188          STRING_PTR[ADDR] = .STRING_PTR[ADDR] + .CURRENT_LEN;
: 4189          STRING_PTR[SIZE] = K_MAX_BUFFER_SIZE - .RET_LEN[0];
```

4152 5066 3 END;
4153 5067 3
4154 5068 4 IF (.SCB[PSMSL_BOTTOM_MARGIN] NEQ 0)
4155 5069 3 THEN BEGIN
4156 5070 4 \$FAO(P
4157 5071 4 BOTTOM_FORMAT,
4158 5072 4 CURRENT_LEN,
4159 5073 4 STRING_PTR[0],
4160 5074 4 | return length
4161 5075 4 SCB[PSMSL_BOTTOM_MARGIN] | address of string
4162 5076 4);
4163 5077 4
4164 5078 4 PUNC_FLAG = 1;
4165 5079 4 RET[EN[0]] = .RET_LEN[0] + .CURRENT_LEN;
4166 5080 4 STRING_PTR[ADDR] = .STRING_PTR[ADDR] + .CURRENT_LEN;
4167 5081 4 STRING_PTR[SIZE] = K_MAX_BUFFER_SIZE - .RET_LEN[0];
4168 5082 3 END;
4169 5083 3
4170 5084 3 IF (.SCB[PSMSL_LEFT_MARGIN] NEQ 0) AND
4171 5085 3 .PUNC_FLAG
4172 5086 3 THEN BEGIN
4173 5087 4 P
4174 5088 4 \$FAO(P
4175 5089 4 PUNCTUATION_FORMAT,
4176 5090 4 CURRENT_LEN,
4177 5091 4 STRING_PTR[0],
4178 5092 4 | return length
4179 5093 4 UPLIT_BYTE (%ASCIC ',') | address of string
4180 5094 4);
4181 5095 4 RET[EN[0]] = .RET_LEN[0] + .CURRENT_LEN;
4182 5096 4 STRING_PTR[ADDR] = .STRING_PTR[ADDR] + .CURRENT_LEN;
4183 5097 3 STRING_PTR[SIZE] = K_MAX_BUFFER_SIZE - .RET_LEN[0];
4184 5098 3 END;
4185 5099 4 IF (.SCB[PSMSL_LEFT_MARGIN] NEQ 0)
4186 5100 3 THEN BEGIN
4187 5101 4 P
4188 5102 4 \$FAO(LEFT_FORMAT,
4189 5103 4 CURRENT[EN],
4190 5104 4 STRING_PTR[0],
4191 5105 4 SCB[PSMSL_LEFT_MARGIN] | return length
4192 5106 4);
4193 5107 4
4194 5108 4 PUNC_FLAG = 1;
4195 5109 4 RET[EN[0]] = .RET_LEN[0] + .CURRENT_LEN;
4196 5110 4 STRING_PTR[ADDR] = .STRING_PTR[ADDR] + .CURRENT_LEN;
4197 5111 4 STRING_PTR[SIZE] = K_MAX_BUFFER_SIZE - .RET_LEN[0];
4198 5112 3 END;
4199 5113 3
4200 5114 3 IF (.SCB[PSMSL_RIGHT_MARGIN] NEQ 0) AND
4201 5115 3 .PUNC_FLAG
4202 5116 3 THEN BEGIN
4203 5117 4 P
4204 5118 4 \$FAO(P
4205 5119 4 PUNCTUATION_FORMAT,
4206 5120 4 CURRENT_LEN,
4207 5121 4 STRING_PTR[0],
4208 5122 4 | return length
4209 5123 4 UPLIT_BYTE (%ASCIC ',') | address of string

```

4209      5123 4      );
4210      5124 4
4211      5125 4      RET_LEN[0] = .RET_LEN[0] + .CURRENT_LEN;
4212      5126 4      STRING_PTR[ADDR] = .STRING_PTR[ADDR] + .CURRENT_LEN;
4213      5127 4      STRING_PTR[SIZE] = K_MAX_BUFFER_SIZE - .RET_LEN[0];
4214      5128 3      END;
4215      5129 3
4216      5130 4      IF (.SCB[PSMSL_RIGHT_MARGIN] NEQ 0)
4217      5131 3      THEN
4218      5132 4      BEGIN
4219      P 5133 4      $FAO( RIGHT FORMAT,
4220      PP 5134 4      CURRENT LEN,
4221      PP 5135 4      STRING PTR[0],           | return length
4222      P 5136 4      .SCB[PSMSL_RIGHT_MARGIN], | address of string
4223      5137 4      );
4224      5138 4
4225      5139 4      RET_LEN[0] = .RET_LEN[0] + .CURRENT_LEN;
4226      5140 4      STRING_PTR[ADDR] = .STRING_PTR[ADDR] + .CURRENT_LEN;
4227      5141 4      STRING_PTR[SIZE] = K_MAX_BUFFER_SIZE - .RET_LEN[0];
4228      5142 3      END;
4229      5143 3
4230      P 5144 3
4231      P 5145 3
4232      P 5146 3
4233      P 5147 3
4234      P 5148 3
4235      5149 3
4236      5150 3
4237      5151 3      RET_LEN[0] = .RET_LEN[0] + .CURRENT_LEN;
4238      5152 3      STRING_PTR[ADDR] = .STRING_PTR[ADDR] + .CURRENT_LEN;
4239      5153 3      STRING_PTR[SIZE] = K_MAX_BUFFER_SIZE - .RET_LEN[0];
4240      5154 2      END;
4241      5155 2
4242      5156 2      IF .RET_LEN[0] GTR K_MAX_BUFFER_SIZE
4243      5157 2      THEN
4244      5158 2      BEGIN
4245      5159 2      RET_LEN[0] = 512;
4246      5160 2      RETURN;
4247      5161 2      END;
4248      5162 2
4249      5163 2      IF NOT (.SBBLOCK[SCB[PSMSL_PRINT_CONTROL], SMBMSG$V_PAGINATE])
4250      5164 2      THEN
4251      5165 2
4252      P 5166 3
4253      P 5167 3
4254      P 5168 3
4255      5169 3
4256      5170 3
4257      5171 3      RET_LEN[0] = .RET_LEN[0] + .CURRENT_LEN;
4258      5172 3      STRING_PTR[ADDR] = .STRING_PTR[ADDR] + .CURRENT_LEN;
4259      5173 3      STRING_PTR[SIZE] = K_MAX_BUFFER_SIZE - .RET_LEN[0];
4260      5174 2      END;
4261      5175 2
4262      5176 2      IF .RET_LEN[0] GTR K_MAX_BUFFER_SIZE
4263      5177 2      THEN
4264      5178 2      BEGIN
4265      5179 3      RET_LEN[0] = 512;

```

```
: 4266      5180 3      RETURN;
: 4267      5181 2      END;
: 4268
: 4269      5183 2      IF (.SCB[PSMSL_FIRST_PAGE] NEQ 0) OR      ! default last page is zero
: 4270      5184 2          (.SCB[PSMSL_LAST_PAGE] NEQ 0)
: 4271      5185 2      THEN
: 4272          BEGIN
: 4273              $FAO ( PAGES FORMAT,
: 4274                  CURRENT LEN,
: 4275                  STRING PTR[0],
: 4276                  .SCB[PSMSL_FIRST_PAGE],
: 4277                  .SCB[PSMSL_LAST_PAGE]
: 4278          );
: 4279
: 4280      5194 3      RET_LEN[0] = .RET_LEN[0] + .CURRENT_LEN;
: 4281      5195 3          STRING_PTR[ADDR] ≡ .STRING_PTR[ADDR] + .CURRENT_LEN;
: 4282      5196 3          STRING_PTR[SIZE] = K_MAX_BUFFER_SIZE - .RET_LEN[0];
: 4283      5197 2      END;
: 4284
: 4285      5199 2      IF .RET_LEN[0] GTR K_MAX_BUFFER_SIZE
: 4286      5200 2      THEN
: 4287          BEGIN
: 4288              RET_LEN[0] = 512;
: 4289              RETURN;
: 4290          END;
: 4291
: 4292      5206 3      IF (.SCB_SIZE_(PARAMETER_1) NEQ 0 OR
: 4293          .SCB_SIZE_(PARAMETER_2) NEQ 0 OR
: 4294          .SCB_SIZE_(PARAMETER_3) NEQ 0 OR
: 4295          .SCB_SIZE_(PARAMETER_4) NEQ 0 OR
: 4296          .SCB_SIZE_(PARAMETER_5) NEQ 0 OR
: 4297          .SCB_SIZE_(PARAMETER_6) NEQ 0 OR
: 4298          .SCB_SIZE_(PARAMETER_7) NEQ 0 OR
: 4299          .SCB_SIZE_(PARAMETER_8) NEQ 0 )
: 4300      5214 2      THEN
: 4301          BEGIN
: 4302              $FAO (
: 4303                  PARAMETER FORMAT,
: 4304                  CURRENT LEN,
: 4305                  STRING PTR[0],
: 4306                  SCB[PSMSQ_PARAMETER_1],
: 4307                  SCB[PSMSQ_PARAMETER_2],
: 4308                  SCB[PSMSQ_PARAMETER_3],
: 4309                  SCB[PSMSQ_PARAMETER_4],
: 4310                  SCB[PSMSQ_PARAMETER_5],
: 4311                  SCB[PSMSQ_PARAMETER_6],
: 4312                  SCB[PSMSQ_PARAMETER_7],
: 4313                  SCB[PSMSQ_PARAMETER_8]
: 4314          );
: 4315
: 4316      5230 3      RET_LEN[0] = .RET_LEN[0] + .CURRENT_LEN;
: 4317      5231 3          STRING_PTR[ADDR] ≡ .STRING_PTR[ADDR] + .CURRENT_LEN;
: 4318      5232 3          STRING_PTR[SIZE] = K_MAX_BUFFER_SIZE - .RET_LEN[0];
: 4319      5233 2      END;
: 4320
: 4321      5235 2      IF .RET_LEN[0] GTR K_MAX_BUFFER_SIZE
: 4322      5236 2      THEN
```

```
: 4323      5237 3      BEGIN
: 4324      5238 3          RET LEN[0] = 512;
: 4325      5239 3          RETURN;
: 4326      5240 2      END;
: 4327      5241 2
: 4328      5242 2      IF .$BBBLOCK[SCB[PSMSL_PRINT_CONTROL], SMBMSG$V_PASSALL]
: 4329      5243 2      THEN
: 4330      5244 2          BEGIN
: 4331      5245 2              $FAO ( PASSALL_FORMAT,
: 4332      5246 2                  CURRENT_LEN
: 4333      5247 2                  STRING_PTR[0]
: 4334      5248 2          );
: 4335      5249 2
: 4336      5250 2          RET LEN[0] = .RET LEN[0] + .CURRENT LEN;
: 4337      5251 2          STRING_PTR[ADDR] = .STRING_PTR[ADDR] + .CURRENT LEN;
: 4338      5252 2          STRING_PTR[SIZE] = K_MAX_BUFFER_SIZE - .RET LEN[0];
: 4339      5253 2      END;
: 4340      5254 2
: 4341      5255 2      IF .RET LEN[0] GTR K_MAX_BUFFER_SIZE
: 4342      5256 2      THEN
: 4343      5257 3          BEGIN
: 4344      5258 3              RET LEN[0] = 512;
: 4345      5259 3              RETURN;
: 4346      5260 2          END;
: 4347      5261 2
: 4348      5262 2      IF .SCB_SIZE_(FILE_SETUP_MODULES) GTR 0
: 4349      5263 2      THEN
: 4350      5264 3          BEGIN
: 4351      5265 3              $FAO ( SETUP FILE FORMAT,
: 4352      5266 3                  CURRENT_LEN
: 4353      5267 3                  STRING_PTR[0]
: 4354      5268 3                  SCB[PSMSQ_FILE_SETUP_MODULES]
: 4355      5269 3          );
: 4356      5270 3
: 4357      5271 3          RET LEN[0] = .RET LEN[0] + .CURRENT LEN;
: 4358      5272 3          STRING_PTR[ADDR] = .STRING_PTR[ADDR] + .CURRENT LEN;
: 4359      5273 3          STRING_PTR[SIZE] = K_MAX_BUFFER_SIZE - .RET LEN[0];
: 4360      5274 2      END;
: 4361      5275 2
: 4362      5276 2      IF .RET LEN[0] GTR K_MAX_BUFFER_SIZE
: 4363      5277 2      THEN
: 4364      5278 3          BEGIN
: 4365      5279 3              RET LEN[0] = 512;
: 4366      5280 3              RETURN;
: 4367      5281 2          END;
: 4368      5282 2
: 4369      5283 2      IF .SCB_SIZE_(FORM_SETUP_MODULES) GTR 0
: 4370      5284 2      THEN
: 4371      5285 3          BEGIN
: 4372      5286 3              $FAO ( SETUP FORM FORMAT,
: 4373      5287 3                  CURRENT_LEN
: 4374      5288 3                  STRING_PTR[0]
: 4375      5289 3                  SCB[PSMSQ_FORM_SETUP_MODULES]
: 4376      5290 3          );
: 4377      5291 3
: 4378      5292 3          RET LEN[0] = .RET LEN[0] + .CURRENT LEN;
: 4379      5293 3          STRING_PTR[ADDR] = .STRING_PTR[ADDR] + .CURRENT LEN;
```

```
: 4380      5294 3     STRING_PTR[SIZE] = K_MAX_BUFFER_SIZE - .RET_LEN[0];
: 4381      5295 2     END;
: 4382      5296 2
: 4383      5297 2     IF .SCB_SIZE_ (PAGE_SETUP_MODULES) GTR 0
: 4384      5298 2     THEN
: 4385      5299 3     BEGIN
: 4386      P 5300 3     $FAO ( SETUP PAGE FORMAT,
: 4387      P 5301 3             CURRENT LEN,
: 4388      P 5302 3             STRING_PTR[0],
: 4389      P 5303 3             SCB[PSMSQ_PAGE_SETUP_MODULES]
: 4390      5304 2         );
: 4391      5305 3
: 4392      5306 3     RET_LEN[0] = .RET_LEN[0] + .CURRENT_LEN;
: 4393      5307 3     STRING_PTR[ADDR] = .STRING_PTR[ADDR] + .CURRENT_LEN;
: 4394      5308 3     STRING_PTR[SIZE] = K_MAX_BUFFER_SIZE - .RET_LEN[0];
: 4395      5309 2     END;
: 4396      5310 2
: 4397      5311 2     IF .RET_LEN[0] GTR K_MAX_BUFFER_SIZE
: 4398      5312 2     THEN
: 4399      5313 3     BEGIN
: 4400      5314 3             RET_LEN[0] = 512;
: 4401      5315 3             RETURN;
: 4402      5316 2     END;
: 4403      5317 2
: 4404      5318 2     IF .$BBLOCK[SCB[PSMSL_PRINT_CONTROL], SMBMSG$V_SHEET_FEED]
: 4405      5319 2     THEN
: 4406      5320 3     BEGIN
: 4407      P 5321 3     $FAO ( SHEET FORMAT,
: 4408      P 5322 3             CURRENT LEN,
: 4409      P 5323 3             STRING_PTR[0]
: 4410      5324 2         );
: 4411      5325 3
: 4412      5326 3     RET_LEN[0] = .RET_LEN[0] + .CURRENT_LEN;
: 4413      5327 3     STRING_PTR[ADDR] = .STRING_PTR[ADDR] + .CURRENT_LEN;
: 4414      5328 3     STRING_PTR[SIZE] = K_MAX_BUFFER_SIZE - .RET_LEN[0];
: 4415      5329 2     END;
: 4416      5330 2
: 4417      5331 2     IF .RET_LEN[0] GTR K_MAX_BUFFER_SIZE
: 4418      5332 2     THEN
: 4419      5333 3     BEGIN
: 4420      5334 3             RET_LEN[0] = 512;
: 4421      5335 3             RETURN;
: 4422      5336 2     END;
: 4423      5337 2
: 4424      5338 2     IF .$BBLOCK[SCB[PSMSL_PRINT_CONTROL], SMBMSG$V_DOUBLE_SPACE]
: 4425      5339 2     THEN
: 4426      5340 3     BEGIN
: 4427      P 5341 3     $FAO ( SPACE FORMAT,
: 4428      P 5342 3             CURRENT LEN,
: 4429      P 5343 3             STRING_PTR[0]
: 4430      5344 2         );
: 4431      5345 3
: 4432      5346 3     RET_LEN[0] = .RET_LEN[0] + .CURRENT_LEN;
: 4433      5347 3     STRING_PTR[ADDR] = .STRING_PTR[ADDR] + .CURRENT_LEN;
: 4434      5348 3     STRING_PTR[SIZE] = K_MAX_BUFFER_SIZE - .RET_LEN[0];
: 4435      5349 2     END;
: 4436      5350 2
```

```
: 4437      5351 2 IF .RET_LEN[0] GTR K_MAX_BUFFER_SIZE
: 4438      5352 2 THEN
: 4439      5353 2 BEGIN
: 4440      5354 2     RET_LEN[0] = 512;
: 4441      5355 2     RETURN;
: 4442      5356 2 END;
: 4443      5357 2 IF .SEPARATE_FLAG_ (FILE_TRAILER)
: 4444      5358 2 THEN
: 4445      5359 2 BEGIN
: 4446      5360 2     SFAO ( TRAILER_FORMAT,
: 4447      5361 2             CURRENT_LEN,
: 4448      5362 2             STRING_PTR[0]
: 4449      5363 2         );
: 4450      5364 2
: 4451      5365 2
: 4452      5366 2     RET_LEN[0] = .RET_LEN[0] + .CURRENT_LEN;
: 4453      5367 2     STRING_PTR[ADDR] = .STRING_PTR[ADDR] + .CURRENT_LEN;
: 4454      5368 2     STRING_PTR[SIZE] = K_MAX_BUFFER_SIZE - .RET_LEN[0];
: 4455      5369 2 END;
: 4456      5370 2 IF .RET_LEN[0] GTR K_MAX_BUFFER_SIZE
: 4457      5371 2 THEN
: 4458      5372 2 BEGIN
: 4459      5373 2     RET_LEN[0] = 512;
: 4460      5374 2     RETURN;
: 4461      5375 2 END;
: 4462      5376 2
: 4463      5377 2 IF .$BBLOCK[SCB[PSMSL_PRINT_CONTROL], SMBMSG$V_TRUNCATE]
: 4464      5378 2 THEN
: 4465      5379 2 BEGIN
: 4466      5380 2     SFAO ( TRUNCATE_FORMAT,
: 4467      5381 2             CURRENTLEN,
: 4468      5382 2             STRING_PTR[0]
: 4469      5383 2         );
: 4470      5384 2
: 4471      5385 2
: 4472      5386 2     RET_LEN[0] = .RET_LEN[0] + .CURRENT_LEN;
: 4473      5387 2     STRING_PTR[ADDR] = .STRING_PTR[ADDR] + .CURRENT_LEN;
: 4474      5388 2     STRING_PTR[SIZE] = K_MAX_BUFFER_SIZE - .RET_LEN[0];
: 4475      5389 2 END;
: 4476      5390 2 IF .RET_LEN[0] GTR K_MAX_BUFFER_SIZE
: 4477      5391 2 THEN
: 4478      5392 2 BEGIN
: 4479      5393 2     RET_LEN[0] = 512;
: 4480      5394 2     RETURN;
: 4481      5395 2 END;
: 4482      5396 2
: 4483      5397 2 IF .RET_LEN[0] GTR K_MAX_BUFFER_SIZE
: 4484      5398 2 THEN
: 4485      5399 2 BEGIN
: 4486      5400 2     SFAO ( WIDTH_FORMAT,
: 4487      5401 2             CURRENT_LEN,
: 4488      5402 2             STRING_PTR[0],
: 4489      5403 2             SCB[PSMSL_FORM_WIDTH]
: 4490      5404 2         );
: 4491      5405 2
: 4492      5406 2     RET_LEN[0] = .RET_LEN[0] + .CURRENT_LEN;
: 4493      5407 2     STRING_PTR[ADDR] = .STRING_PTR[ADDR] + .CURRENT_LEN;
: 4494      5408 2     STRING_PTR[SIZE] = K_MAX_BUFFER_SIZE - .RET_LEN[0];
```

```

: 4494      5408 2 IF .RET_LEN[0] GTR K_MAX_BUFFER_SIZE
: 4495      5409 2 THEN
: 4496      5410 3 BEGIN
: 4497      5411 3   RET_LEN[0] = 512;
: 4498      5412 3   RETURN;
: 4499      5413 2 END;
: 4500      5414 2
: 4501      5415 2 IF .$BBBLOCK[SCB[PSMSL_PRINT_CONTROL], SMBMSGV_WRAP]
: 4502      5416 2 THEN
: 4503      5417 3 BEGIN
: 4504      P 5418 3   $FAO ( WRAP_FORMAT,
: 4505      P 5419 3   CURRENT_LEN,
: 4506      P 5420 3   STRING_PTR[0] ) ;                                ! return length
: 4507      5421 3
: 4508      5422 3
: 4509      5423 3   RET_LEN[0] = .RET_LEN[0] + .CURRENT_LEN;
: 4510      5424 2 END;
: 4511      5425 2
: 4512      5426 2 ! Don't print anything if no flags were set
: 4513      5427 2
: 4514      5428 2 IF .RET_LEN[0] LEQ 18
: 4515      5429 2 THEN
: 4516      5430 2   RET_LEN[0] = 0;
: 4517      5431 2
: 4518      5432 2 ! Length returned must be less than max string size
: 4519      5433 2
: 4520      5434 2 IF .RET_LEN[0] GTR K_MAX_BUFFER_SIZE
: 4521      5435 2 THEN
: 4522      5436 3 BEGIN
: 4523      5437 3   RET_LEN[0] = 512;
: 4524      5438 3   RETURN;
: 4525      5439 2 END;
: 4526      5440 2
: 4527      5441 1 END;

```

44 25 37 31 21 3D 52 45 54 46 41 2F 20	01DA3 P.ADB: .ASCII \!17%D\
	00000005 01DA8 P.ADA: .LONG 5
	00000000 01DAC P.ADD: .ADDRESS P.ADB
54 53 52 55 42 2F 20	01DB0 P.ADD: .ASCII \ /AFTER=!17%D\
	0000000D 01DBD P.ADC: .BLKB 3
	00000000 01DC0 P.ADC: .LONG 13
54 53 52 55 42 2F 20	01DC4 P.ADF: .ADDRESS P.ADD
	00000007 01DC8 P.ADF: .ASCII \ /BURST\
	00000000 01DCF P.ADE: .BLKB 1
49 54 53 49 52 45 54 43 41 52 41 48 43 2F 20	01DD0 P.ADE: .LONG 7
	00000007 01DD4 P.ADH: .ADDRESS P.ADF
	00000000 01DD8 P.ADH: .ASCII \ /CHARACTERISTICS=!AS\
55 21 28 4C 55 21 3D 53 45 49 50 4F 43 2F 20	01DE7 P.ADH: .BLKB 3
	00000015 01DF0 P.ADG: .LONG 21
	00000000 01DF4 P.ADG: .ADDRESS P.ADH
29 4C 55 21 20 66 6F 20 4C	01DF8 P.ADJ: .ASCII \ /COPIES=!UL(!UL of !UL)\
	00000018 01E07 P.ADJ: .LONG 24
	00000000 01E10 P.ADI: .ADDRESS P.ADJ
	00000000 01E14 P.ADI: .LONG 24

SEPARATE
V04-001Print Symbiont -- separation routines
GET_QUALIFIERS - Get Switches/Qualifiers associN 7
16-Sep-1984 02:23:03 VAX-11 Bliss-32 v4.0-742
14-Sep-1984 22:32:26 [PRTSMB.SRC]SEPARATE.B32;2Page 139
(34)

44	45	45	46	2F	20	01E18	P.ADl:	.ASCII \ /FEED\
				00000006.		01E1E	.BLKB	2
				00000000.		01E20	P.ADK:	.LONG 6
				00000006.		01E24	.ADDRESS P.ADl	
				00000000.		01E28	P.ADN:	.ASCII \ /FLAG\
				00000006.		01E2E	.BLKB	2
				00000000.		01E30	P.ADM:	.LONG 6
				00000006.		01E34	.ADDRESS P.ADN	
				00000000.		01E38	P.ADP:	.ASCII \ /FORM=!AS\
				0000000A.		01E42	.BLKB	2
				00000000.		01E44	P.ADO:	.LONG 10
				00000008.		01E48	.ADDRESS P.ADP	
				00000000.		01E4C	P.ADR:	.ASCII \ /HEADER\
				00000008.		01E54	P.ADQ:	.LONG 8
				00000000.		01E58	.ADDRESS P.ADR	
				00000000.		01E5C	P.ADT:	.ASCII \ /JOB_COUNT=!UL(!UL of !UL)\
				0000001B.		01E6B		
				00000000.		01E77	.BLKB	1
				00000000.		01E78	P.ADS:	.LONG 27
				0000000C.		01E7C	.ADDRESS P.ADT	
				00000000.		01E80	P.ADV:	.ASCII \ /LENGTH=!UL\
				00000000.		01E8C	P.ADU:	.LONG 12
				00000000.		01E90	.ADDRESS P.ADV	
				0000000D.		01E94	P.ADX:	.ASCII \ /LIBRARY=!AS\
				00000000.		01EA1	.BLKB	3
				0000000A.		01EA4	P.ADW:	.LONG 13
				00000000.		01EA8	.ADDRESS P.ADX	
				00000000.		01EAC	P.ADZ:	.ASCII \ /MARGIN=(\
				00000000.		01EB6	.BLKB	2
				0000000A.		01EBC	P.ADY:	.LONG 10
				00000000.		01EC0	P.AEB:	.ASCII \ /TOP=!UL\
				00000000.		01EC7	.BLKB	1
				00000007.		01EC8	P.AEA:	.LONG 7
				00000000.		01ECC	.ADDRESS P.AEB	
				0000000A.		01ED0	P.AED:	.ASCII \ /BOTTOM=!UL\
				00000000.		01EDA	.BLKB	2
				00000008.		01EDC	P.AEC:	.LONG 10
				00000000.		01EE0	.ADDRESS P.AED	
				0000004C.		01EE4	P.AEF:	.ASCII \ /LEFT=!UL\
				00000008.		01EEC	P.AEE:	.LONG 8
				00000000.		01EFO	.ADDRESS P.AEF	
				00000000.		01EF4	P.AEH:	.ASCII \ /RIGHT=!UL\
				00000009.		01EFD	.BLKB	3
				00000000.		01F00	P.AEG:	.LONG 9
				00000008.		01F04	.ADDRESS P.AEH	
				00000000.		01F08	P.AEJ:	.ASCII \ /NOFEED\
				00000008.		01F10	P.AEI:	.LONG 8
				00000000.		01F14	.ADDRESS P.AEJ	
				00000000.		01F18	P.AEL:	.ASCII \ /PAGE_SETUP=(!AS)\
				00000012.		01F27		
				00000000.		01F2A	.BLKB	2
				00000000.		01F2C	P.AEK:	.LONG 18
				00000000.		01F30	.ADDRESS P.AEL	
				00000000.		01F34	P.AEN:	.ASCII \ /PAGES=(!UL,!UL)\
				00000000.		01F43		
				00000000.		01F45	.BLKB	3

SEPARATE
V04-001

Print Symbiont -- separation routines
GET_QUALIFIERS - Get Switches/Qualifie

B 8
16-Sep-1984 02:23:03
14-Sep-1984 22:32:26

VAX-11 Bliss-32 V4.0-742
[PRTSMB.SRC]SEPARATE.B32;2

Page 140
(34)

SEP
V04

DATE_FORMAT= P.ADA
 AFTER_TIME_FORMAT= P.ADC
 BURST_FORMAT= P.ADE
 CHARACTERISTICS_FORMAT= P.ADG
 FILE_COPIES_FORMAT= P.ADI
 FEED_FORMAT= P.ADK
 FLAG_FORMAT= P.ADM
 FORM_FORMAT= P.ADO
 HEADER_FORMAT= P.ADQ
 JOB_COUNT_FORMAT= P.ADS
 LENGTH_FORMAT= P.ADU
 LIBRARY_FORMAT= P.ADW
 MARGIN_FORMAT= P.ADY
 TOP_FORMAT= P.AEA
 BOTTOM_FORMAT= P.AEC
 LEFT_FORMAT= P.AEE
 RIGHT_FORMAT= P.AEG
 NOFEED_FORMAT= P.AEI
 SETUP_PAGE_FORMAT= P.AEK
 PAGES_FORMAT= P.AEM
 PARAMETER_FORMAT= P.AEO
 PASSALL_FORMAT= P.AEQ
 PUNCTUATION_FORMAT= P.AES
 SETUP_FILE_FORMAT= P.AEU
 SETUP_FORM_FORMAT= P.AEW
 SHEET_FORMAT= P.AEY
 SPACE_FORMAT= P.AFA
 TRAILER_FORMAT= P.AFC
 TRUNCATE_FORMAT= P.AFE
 WIDTH_FORMAT= P.AFG
 WRAP_FORMAT= P.AFI

07FC 00000 GET_QUALIFIERS:

				.WORD	Save R2,R3,R4,R5,R6,R7,R8,R9,R10	: 4690
5A	FF42	CF	9E 00002	MOVAB	PUNCTUATION_FORMAT, R10	
59	00000000G	00	9E 00007	MOVAB	SYSSFAO, R9	
5E	AC	AE	9E 0000E	MOVAB	-84(SP), SP	
		56	D4 00012	CLRL	PUNC FLAG	: 4695
14	AE	08	AE D4 00014	CLRL	CURRENT LEN	
		0200	BF 3C 00017	MOVZWL	#512, STRING_PTR	: 4810
18	50	08	AC D0 0001D	MOVL	STR DESC, R0	: 4811
	AE	04	A0 D0 00021	MOVL	4(R0), STRING_PTR+4	
	55	0C	AC D0 00026	MOVL	RET LEN, R5	: 4813
			65 B4 0002A	CLRW	(R5)	
4C	AE	11	D0 0002C	MOVL	#17, AFT_DATE_PTR	: 4827
50	AE	30	AE 9E 00030	MOVAB	AFT_BUFF, AFT_DATE_PTR+4	: 4828
	54	04	AC D0 00035	MOVL	SCB, R4	: 4833
		24	A4 9F 00039	PUSHAB	36(R4)	
		50	AE 9F 0003C	PUSHAB	AFT_DATE_PTR	
		OC	AE 9F 0003F	PUSHAB	TEMP_LEN	
		FDF0	CA 9F 00042	PUSHAB	DATE_FORMAT	
44	69	04	FB 00046	CALLS	#4, SYSSFAO	
48	AE	11	D0 00049	MOVL	#17, TEMP_PTR	: 4835
		1C	AE 9E 0004D	MOVAB	TEMP_BUFF, TEMP_PTR+4	: 4836
		015C	C4 9F 00052	PUSHAB	348(R4)	: 4841

					PUSHAB TEMP_PTR	
					PUSHAB TEMP_LEN	
					PUSHAB DATE_FORMAT	
					CALLS #4, SYSSFAO	
					CMPC3 TEMP_LEN, @TEMP_PTR+4, @AFT_DATE_PTR+4	
					BEQL 1\$	4843
					PUSHAB 36(R4)	4851
					PUSHAB STRING_PTR	
					PUSHAB CURRENT_LEN	
					PUSHAB AFTER_TIME_FORMAT	
					CALLS #4, SYSSFAO	
					ADDW2 CURRENT_LEN, (R5)	4853
					ADDL2 CURRENT_LEN, STRING_PTR+4	4854
					MOVZWL (R5), STRING_PTR	4855
					SUBL3 STRING_PTR, #512, STRING_PTR	
					MOVAB 340(R4), R8	4858
					BLBC (R8), 2\$	
					PUSHAB STRING_PTR	4864
					PUSHAB CURRENT_LEN	
					PUSHAB BURST_FORMAT	
					CALLS #3, SYSSFAO	
					ADDW2 CURRENT_LEN, (R5)	4866
					ADDL2 CURRENT_LEN, STRING_PTR+4	4867
					MOVZWL (R5), STRING_PTR	4868
					SUBL3 STRING_PTR, #512, STRING_PTR	
					TSTW 52(R4)	4871
					BNEQ 3\$	
					PUSHAB 52(R4)	4878
					PUSHAB STRING_PTR	
					PUSHAB CURRENT_LEN	
					PUSHAB CHARACTERISTICS_FORMAT	
					CALLS #4, SYSSFAO	
					ADDW2 CURRENT_LEN, (R5)	4880
					ADDL2 CURRENT_LEN, STRING_PTR+4	4881
					MOVZWL (R5), STRING_PTR	4882
					SUBL3 STRING_PTR, #512, STRING_PTR	
					MOVAB 292(R4), R7	4887
					#2, (R7), 4\$	
					PUSHAB STRING_PTR	4893
					PUSHAB CURRENT_LEN	
					PUSHAB FEED_FORMAT	
					CALLS #3, SYSSFAO	
					ADDW2 CURRENT_LEN, (R5)	4895
					ADDL2 CURRENT_LEN, STRING_PTR+4	4896
					MOVZWL (R5), STRING_PTR	4897
					SUBL3 STRING_PTR, #512, STRING_PTR	
					CMPL 100(R4), #1	4900
					BLEQ 5\$	
					PUSHL 100(R4)	4909
					MOVL 100(R4), -(SP)	
					PUSHAB STRING_PTR	
					PUSHAB CURRENT_LEN	
					PUSHAB FILE_COPIES_FORMAT	
					CALLS #6, SYSSFAO	
					ADDW2 CURRENT_LEN, (R5)	4911
					ADDL2 CURRENT_LEN, STRING_PTR+4	4912
					MOVZWL (R5), STRING_PTR	4913

SEPARATE
V04-001

Print Symbiont -- separation routines
GET_QUALIFIERS - Get Switches/Qualifie

E B
16-Sep-1984 02:23:03
14-Sep-1984 22:32:26

VAX-11 Bliss-32 V4.0-742
[PRTSMB.SRC]SEPARATE.B32:2

Page 143
(34)

SEP
V04

SEPARATE
V04-001

Print Symbiont -- separation routines
GET_QUALIFIERS - Get Switches/Qualifier

F 8
16-Sep-1984 02:23:03
14-Sep-1984 22:32:26

VAX-11 Bliss-32 V4.0-742
[PRTSMB.SRC]SEPARATE.B32:2

Page 144
(34)

SEF
V04

SEPARATE
V04-001

Print Symbiont -- separation routines
GET_QUALIFIERS - Get Switches/Qualifi

6 8
16-Sep-1984 02:23:03
14-Sep-1984 22:32:26

VAX-11 Bliss-32 v4.0-742
[PRTSMB.SRC]SEPARATE.B32;2

Page 145
(34)

SEPARATE
V04-001

Print_Symbiont -- separation routines
GET_QUALIFIERS - Get Switches/Qualifiers

H 8
16-Sep-1984 02:23:03
14-Sep-1984 22:32:26

VAX-11 Bliss-32 V4.0-742
[PRTSMB.SRC]SEPARATE.B32:2

Page 146
(34)

SEP
V04

14	AE 00000200	14	AE 8F	14	65 AE	3C C3	003DB 003DF	MOVZWL SUBL3	(R5), STRING PTR STRING PTR, #512, STRING_PTR
			26		53	E9	003E9	BLBC	R3, 20\$
				18	52 AE	9F	003EE	PUSHL	R2
				10	AE 9F	003F1		PUSHAB	STRING PTR
				FF48	CA 9F	003F4		PUSHAB	CURRENT LEN
			69	04	FB	003F8		PUSHAB	RIGHT FORMAT
			65	08 AE	A0	003FB	CALLS	#4, SYSSFAO	
			18	08	AE CO	003FF	ADDW2	CURRENT_LEN, (R5)	
			14	AE	65	3C	00404	ADDL2	CURRENT_LEN, STRING_PTR+4
14	AE 00000200	14	AE 8F	14	AE	C3	00408	MOVZWL	(R5), STRING PTR
			00B6	CA	9F	00412	SUBL3	STRING_PTR, #512, STRING_PTR	
				18	AE 9F	00416	PUSHAB	P.AFN	
				10	AE 9F	00419	PUSHAB	STRING PTR	
			69	5A	DD	0041C	PUSHAB	CURRENT_LEN	
			65	04	FB	0041E	PUSHL	R10	
			18	08 AE	A0	00421	CALLS	#4, SYSSFAO	
			14	08	AE CO	00425	ADDW2	CURRENT_LEN, (R5)	
14	AE 00000200	18	AE 8F	14	AE	C3	0042E	ADDL2	CURRENT_LEN, STRING_PTR+4
		14	02	65	B1	00438	MOVZWL	(R5), STRING PTR	
			69	1A	0043D	SUBL3	STRING_PTR, #512, STRING_PTR		
			0C	AE 9F	00443	CMPW	(R5), #512		
			FF58	CA	9F	00446	BGTRU	25\$	
			69	03	FB	00449	BBS	#2, (R7), 22\$	
			65	08 AE	A0	00450	PUSHAB	STRING PTR	
			18	08	AE CO	00454	PUSHAB	CURRENT LEN	
			14	AE	65	3C	00459	PUSHAB	NOFEED FORMAT
14	AE 00000200	14	AE 8F	14	AE	C3	0045D	CALLS	#3, SYSSFAO
			00B8	65	B1	00467	ADDW2	CURRENT_LEN, (R5)	
				74	A4	D5	0046C	ADDL2	CURRENT_LEN, STRING_PTR+4
				06	12	00471	MOVZWL	(R5), STRING PTR	
			00B8	C4	D5	00473	SUBL3	STRING_PTR, #512, STRING_PTR	
				2A	13	00477	CMPW	(R5), #512	
			00B8	C4	DD	00479	BGTRU	25\$	
				74	A4	DD	0047D	TSTL	116(R4)
				1C	AE	9F	00480	BNEQ	23\$
				14	AE	9F	00483	TSTL	184(R4)
				90	AA	9F	00486	BEQL	24\$
			69	05	FB	00489	PUSHL	184(R4)	
			18	65	08 AE	A0	0048C	PUSHL	116(R4)
			14	08	AE	CO	00490	PUSHAB	STRING PTR
14	AE 00000200	14	AE 8F	14	AE	C3	00499	PUSHAB	CURRENT LEN
			00E4	65	B1	004A3	PUSHAB	PAGES FORMAT	
				78	1A	004A8	CALLS	#5, SYSSFAO	
			00EC	C4	B5	004AA	ADDW2	CURRENT_LEN, (R5)	
				2A	12	004AE	ADDL2	CURRENT_LEN, STRING_PTR+4	
			00F4	C4	B5	004B0	MOVZWL	(R5), STRING PTR	
				24	12	004B4	SUBL3	STRING_PTR, #512, STRING_PTR	
			00FC	C4	B5	004B6	CMPW	(R5), #512	
				1E	12	004BA	BGTRU	28\$	
			00E4	C4	B5	004BC	TSTW	228(R4)	
				2A	12	004AE	BNEQ	26\$	
			00EC	C4	B5	004B0	TSTW	236(R4)	
				24	12	004B4	BNEQ	26\$	
			00F4	C4	B5	004B6	TSTW	244(R4)	
				1E	12	004BA	BNEQ	26\$	
			00FC	C4	B5	004BC	TSTW	252(R4)	

**SEPARATE
V04-001**

Print_Symbiont -- separation routines
GET_QUALIFIERS - Get Switches/Qualifier

J 8
16-Sep-1984 02:23:03
14-Sep-1984 22:32:26

VAX-11 Bliss-32 V4.0-742
[PRTSMB.SRC]SEPARATE.B32:2

Page 148
(34)

SEPARATE
V04-001

Print Symbiont -- separation routines
GET_QUALIFIERS - Get Switches/Qualifiers

K 8
16-Sep-1984 02:23:03
14-Sep-1984 22:32:26

VAX-11 BLISS-32 V4.0-742
[PRTSMB.SRC]SEPARATE.B32:2

Page 149
(34)

SEF
V04

14	AE 00000200 0200	18 14 AE 8F 008C 10 0098 69 65 18 14 AE AE 08 AE 65 3C 006B1 006B6 C3 006BA B1 006C4 1A 006C9 67 95 006CB 11 18 006CD 14 0C 00A8 69 65 12 0200	08 14 AE 8F 008C 18 10 0098 69 65 08 AE 65 3C 006B1 006B6 C3 006BA B1 006C4 23 1A 006C9 67 95 006CB 11 18 006CD 14 0C 00A8 03 FB 006D9 AE A0 006DC B1 006E0 02 1A 006E3 65 B4 006E5 65 B1 006E7 05 1B 006EC 8F B0 006EE 04 006F3	CO 3C C3 B1 52 1A DD AE 9F AE 9F CA 04 FB AE A0 AE C0 3C C3 B1 23 1A 006C9 67 95 006CB 11 18 006CD 14 0C 00A8 03 FB 006D9 AE A0 006DC B1 006E0 02 1A 006E3 65 B4 006E5 65 B1 006E7 05 1B 006EC 8F B0 006EE 04 006F3	00682 00687 0068B 00695 0069A 0069C 006A0 006A3 006A6 006AA 006AD 006B1 006B6 006BA 006C4 006C9 006CB 006CD 006CF 006D2 006D5 006D9 006DC 006E0 006E3 006E5 006E7 006EC 006EE 006F3	00682 00687 0068B 00695 0069A 0069C 006A0 006A3 006A6 006AA 006AD 006B1 006B6 006BA 006C4 006C9 006CB 006CD 006CF 006D2 006D5 006D9 006DC 006E0 006E3 006E5 006E7 006EC 006EE 006F3	40\$: 41\$: 42\$: 43\$: 44\$: 45\$: RET	ADDL2 MOVZWL SUBL3 CMPW BGTRU PUSHL PUSHAB PUSHAB PUSHAB CALLS ADDW2 ADDL2 MOVZWL SUBL3 CMPW BGTRU TSTB BGEQ PUSHAB PUSHAB PUSHAB CALLS ADDW2 CMPW BGTRU CLRW CMPW BLEQU MOVW RET	CURRENT_LEN, STRING_PTR+4 (RS), STRING_PTR STRING_PTR, #512, STRING_PTR (R5), #512 45\$ 140(R4) STRING_PTR CURRENT_LEN WIDTH_FORMAT #4, SYSSFAO CURRENT_LEN, (R5) CURRENT_LEN, STRING_PTR+4 (RS), STRING_PTR STRING_PTR, #512, STRING_PTR (R5), #512 45\$ (R7) 43\$ STRING_PTR CURRENT_LEN WRAP_FORMAT #3, SYSSFAO CURRENT_LEN, (R5) (R5), #T8 44\$ (R5) (R5), #512 46\$ #512, (R5)

; Routine Size: 1780 bytes, Routine Base: CODE + 2070

5429 : 542 1 %sbttl 'GET_QUEUE_QUALIFIERS - Gets the qualifiers pertaining to queues'
5430 : 543 1 ++
5431 : 544 1 Functional Description:
5432 : 545 1 This routine returns a string containing the all relevant file qualifier
5433 : 546 1 information.
5434 : 547 1 Formal Parameters:
5435 : 548 1 SCB - Address of the SCB
5436 : 549 1 STR_DESC - Desc of String to Return
5437 : 550 1 RET_LEN - Return length of Desc.
5438 :
5439 :
5440 : 553 1 Implicit Inputs:
5441 : 554 1 none
5442 : 555 1 Implicit Outputs:
5443 : 556 1 none
5444 : 557 1 Returned Value:
5445 : 558 1 none
5446 : 559 1 Side Effects:
5447 : 560 1 none
5448 : 561 1 --
5449 : 562 1 ROUTINE GET_QUEUE_QUALIFIERS (! SCB
5450 : 563 1 SCB : REF \$BLOCK, ! Output buffer desc
5451 : 564 1 STR_DESC : REF VECTOR[2], ! Return length (word)
5452 : 565 1 RET_LEN : REF VECTOR [,WORD]
5453 : 566 1) : NOVALUE =
5454 :
5455 :
5456 :
5457 :
5458 :
5459 : 567 1 BEGIN
5460 : 568 1 BIND
5461 : P 569 1 BEGIN FORMAT = \$DESCRIPTOR (! - Reset Module
5462 : 570 1 'Queue Qualifiers:').
5463 : P 571 1 RESET FORMAT = \$DESCRIPTOR (! - separation
5464 : 572 1 'RESET=""',
5465 : 573 1 '!AS',
5466 : 574 1 '""').
5467 : P 575 1 JOB_RESET_MODULE FORMAT = \$DESCRIPTOR (! - separation
5468 : 576 1 '/SEPARATE=T').
5469 : P 577 1 INSERTION FORMAT = \$DESCRIPTOR (! - separation flags
5470 : 578 1 '!AC').
5471 :
5472 :
5473 :
5474 : 579 1 LITERAL
5475 : 580 1 K_MAX_BUFFER_SIZE = 512;
5476 :
5477 : 581 1 LOCAL
5478 : 582 1 INSERT_FLAG : INITIAL (0), ! Pointer to current string
5479 : 583 1 CURRENT_LEN : INITIAL (0),
5480 : 584 1 STRING_PTR : VECTOR [2];
5481 :
5482 : 585 1 STRING_PTR[SIZE] = K_MAX_BUFFER_SIZE;
5483 : 586 1 STRING_PTR[ADDR] = .STR_DESC[ADDR]; ! init address
5484 :
5485 :
5486 :
5487 : 587 1 RET_LEN[0] = 0;

```

: 4586
: 4587
: 4588
: 4589
: 4590
: 4591
: 4592
: 4593
: 4594
: 4595
: 4596
: 4597
: 4598
: 4599
: 4600
: 4601
: 4602
: 4603
: 4604
: 4605
: 4606
: 4607
: 4608
: 4609
: 4610
: 4611
: 4612
: 4613
: 4614
: 4615
: 4616
: 4617
: 4618
: 4619
: 4620
: 4621
: 4622
: 4623
: 4624
: 4625
: 4626
: 4627
: 4628
: 4629
: 4630
: 4631
: 4632
: 4633
: 4634
: 4635
: 4636
: 4637
: 4638
: 4639
: 4640
: 4641
: 4642

5499 2
P 5500 2 SFAO ( BEGIN FORMAT,
P 5501 2 CURRENT LEN,
P 5502 2 STRING_PTR[0],
P 5503 2 );
P 5504 2
P 5505 2 RET_LEN[0] = .RET_LEN[0] + .CURRENT_LEN;
P 5506 2 STRING_PTR[ADDR] ≡ .STRING_PTR[ADDR] + .CURRENT_LEN;
P 5507 2 STRING_PTR[SIZE] = K_MAX_BUFFER_SIZE - .RET_LEN[0];
P 5508 2
P 5509 2
P 5510 2 IF .SEPARATE FLAG_ (JOB_BURST) OR
P 5511 2 .SEPARATE_FLAG_ (JOB_FLAG) OR
P 5512 2 .SEPARATE_FLAG_ (JOB_TRAILER) OR
P 5513 2 .SCB_SIZE_ (JOB_RESET_MODULES)
P 5514 2 THEN
D 5515 2 BEGIN
P 5516 2 SFAO ( JOB_RESET_MODULE_FORMAT,
P 5517 2 CURRENT LEN,
P 5518 2 STRING_PTR[0],
P 5519 2 );
P 5520 2
P 5521 2 RET_LEN[0] = .RET_LEN[0] + .CURRENT_LEN;
P 5522 2 STRING_PTR[ADDR] ≡ .STRING_PTR[ADDR] + .CURRENT_LEN;
P 5523 2 STRING_PTR[SIZE] = K_MAX_BUFFER_SIZE - .RET_LEN[0];
P 5524 2
P 5525 3 IF .SEPARATE_FLAG_ (JOB_BURST)
P 5526 3 THEN
P 5527 4 BEGIN
P 5528 4 SFAO (
P 5529 4 INSERTION FORMAT,
P 5530 4 CURRENT LEN,
P 5531 4 STRING_PTR[0],
P 5532 4 UPLIT BYTE (%ASCIC'BURST')
P 5533 4 );
P 5534 4
P 5535 4 INSERT_FLAG = 1;
P 5536 4 RET_LEN[0] = .RET_LEN[0] + .CURRENT_LEN;
P 5537 4 STRING_PTR[ADDR] ≡ .STRING_PTR[ADDR] + .CURRENT_LEN;
P 5538 4 STRING_PTR[SIZE] = K_MAX_BUFFER_SIZE - .RET_LEN[0];
P 5539 4 END;
P 5540 3
P 5541 3 IF .SEPARATE_FLAG_ (JOB_FLAG) AND
P 5542 3 .INSERT_FLAG
P 5543 3 THEN
P 5544 4 BEGIN
P 5545 4 SFAO (
P 5546 4 INSERTION FORMAT,
P 5547 4 CURRENT LEN,
P 5548 4 STRING_PTR[0],
P 5549 4 UPLIT BYTE (%ASCIC',')
P 5550 4 );
P 5551 4
P 5552 4 RET_LEN[0] = .RET_LEN[0] + .CURRENT_LEN;
P 5553 4 STRING_PTR[ADDR] ≡ .STRING_PTR[ADDR] + .CURRENT_LEN;
P 5554 4 STRING_PTR[SIZE] = K_MAX_BUFFER_SIZE - .RET_LEN[0];
P 5555 3

```

4643 5556 3
4644 5557 3
4645 5558 3
4646 5559 4
4647 P 5560 4
4648 P 5561 4
4649 P 5562 4
4650 P 5563 4
4651 P 5564 4
4652 5565 4
4653 5566 4
4654 5567 4
4655 5568 4
4656 5569 4
4657 5570 4
4658 5571 4
4659 5572 4
4660 5573 4
4661 5574 4
4662 5575 4
4663 5576 4
4664 P 5577 4
4665 P 5578 4
4666 P 5579 4
4667 P 5580 4
4668 P 5581 4
4669 5582 4
4670 5583 4
4671 5584 4
4672 5585 4
4673 5586 4
4674 5587 4
4675 5588 4
4676 5589 4
4677 5590 4
4678 5591 4
4679 P 5592 4
4680 P 5593 4
4681 P 5594 4
4682 P 5595 4
4683 P 5596 4
4684 5597 4
4685 5598 4
4686 5599 4
4687 5600 4
4688 5601 4
4689 5602 4
4690 5603 4
4691 5604 4
4692 5605 4
4693 5606 4
4694 5607 4
4695 P 5608 4
4696 P 5609 4
4697 P 5610 4
4698 P 5611 4
4699 P 5612 4

IF .SEPARATE_FLAG_ (JOB_FLAG)
THEN
BEGIN
\$FAO (
INSERTION FORMAT,
CURRENT LEN,
STRING PTR[0],
UPLIT BYTE (%ASCIC'FLAG')
);

INSERT FLAG = 1;
RET_LEN[0] = .RET_LEN[0] + .CURRENT LEN;
STRING_PTR[ADDR] = .STRING_PTR[ADDR] + .CURRENT LEN;
STRING_PTR[SIZE] = K_MAX_BUFFER_SIZE - .RET_LEN[0];
END;

IF .SCB_SIZE_(JOB_RESET_MODULES) AND
.INSERT_FLAG
THEN
BEGIN
\$FAO (
INSERTION FORMAT,
CURRENT LEN,
STRING PTR[0],
UPLIT BYTE (%ASCIC',')
);

RET_LEN[0] = .RET_LEN[0] + .CURRENT LEN;
STRING_PTR[ADDR] = .STRING_PTR[ADDR] + .CURRENT LEN;
STRING_PTR[SIZE] = K_MAX_BUFFER_SIZE - .RET_LEN[0];
END;

IF .SCB_SIZE_(JOB_RESET_MODULES)
THEN
BEGIN
\$FAO (
RESET FORMAT,
CURRENT LEN,
STRING PTR[0],
SCB[PSMSQ_JOB_RESET_MODULES]
);

RET_LEN[0] = .RET_LEN[0] + .CURRENT LEN;
STRING_PTR[ADDR] = .STRING_PTR[ADDR] + .CURRENT LEN;
STRING_PTR[SIZE] = K_MAX_BUFFER_SIZE - .RET_LEN[0];
END;

IF .SEPARATE_FLAG_ (JOB_TRAILER) AND
.INSERT_FLAG
THEN
BEGIN
\$FAO (

INSERTION FORMAT,
CURRENT LEN,
STRING PTR[0],
UPLIT BYTE (%ASCIC',')

```

4700      5613 4   );
4701      5614 4
4702      5615 4   RET_LEN[0] = .RET_LEN[0] + .CURRENT_LEN;
4703      5616 4   STRING_PTR[ADDR] = .STRING_PTR[ADDR] + .CURRENT_LEN;
4704      5617 4   STRING_PTR[SIZE] = K_MAX_BUFFER_SIZE - .RET_LEN[0];
4705      5618 3   END;
4706      5619 3
4707      5620 3
4708      5621 3
4709      5622 4   IF .SEPARATE_FLAG_ (JOB_TRAILER)
4710      P 5623 4   THEN
4711      P 5624 4   BEGIN
4712      P 5625 4   SFAO (
4713      P 5626 4   INSERTION FORMAT,
4714      P 5627 4   CURRENT LEN
4715      P 5628 4   STRING_PTR[0],
4716      P 5629 4   UPLIT BYTE (%ASCIC' TRAILER')
4717      P 5630 4   );
4718      P 5631 4   RET_LEN[0] = .RET_LEN[0] + .CURRENT_LEN;
4719      P 5632 4   STRING_PTR[ADDR] = .STRING_PTR[ADDR] + .CURRENT_LEN;
4720      P 5633 3   STRING_PTR[SIZE] = K_MAX_BUFFER_SIZE - .RET_LEN[0];
4721      P 5634 3   END;
4722      P 5635 3
4723      P 5636 3
4724      P 5637 3
4725      P 5638 3
4726      P 5639 3   SFAO (
4727      P 5640 3   INSERTION FORMAT,
4728      P 5641 3   CURRENT LEN
4729      P 5642 3   STRING_PTR[0],
4730      P 5643 3   UPLIT BYTE (%ASCIC') )
4731      P 5644 3
4732      P 5645 2   );
4733      P 5646 2
4734      P 5647 2   ! Don't print anything if no flags were set
4735      P 5648 2
4736      P 5649 2   IF .RET_LEN[0] LEQ 18
4737      P 5650 2   THEN
4738      P 5651 2   RET_LEN[0] = 0;
4739      P 5652 2
4740      P 5653 2   ! Length returned must be less than max string size
4741      P 5654 2   IF .RET_LEN[0] GTR K_MAX_BUFFER_SIZE
4742      P 5655 2   THEN
4743      P 5656 3   BEGIN
4744      P 5657 3   RET_LEN[0] = 512;
4745      P 5658 3   RETURN;
4746      P 5659 2   END;
4747      P 5660 2
4748      P 5661 1 END;

```

72 65 69 66 69 6C 61 75 51 20 65 75 65 75 51 02764 P.AFP: .ASCII \Queue Qualifiers:\
 3A 73 02773
 02775
 00000011 02778 P.AFO: .BLKB 3
 00000000 0277C .LONG 17
 .ADDRESS P.AFP

	22	3D	54	45	53	45	52	02780	P.AFR:	.ASCII	\RESET='\'	
					53	41	21	02787		.ASCII	\!AS\	
							22	0278A		.ASCII	\\"	
								0278B		.BLKB	1	
								0278C	P.AFQ:	.LONG	11	
28	3D	45	54	41	52	41	50	45	02790	.ADDRESS	P.AFR	
					53	2F	20	02794	P.AFT:	.ASCII	\/SEPARATE=(\	
						00000000C.		027A0	P.AFS:	.LONG	12	
						00000000.		027A4	.ADDRESS	P.AFT		
						43	41	21	027A8	P.AFV:	.ASCII	\!AC\
								027AB		.BLKB	1	
								027AC	P.AFU:	.LONG	3	
								027B0	.ADDRESS	P.AFV		
	54	53	52	55	42	05		027B4	P.AFW:	.ASCII	<5>\BURST\	
						2C	01	027BA	PAFX:	.ASCII	<1>_\	
				47	41	4C	46	04	027BC	P.AFY:	.ASCII	<4>\FLAG\
							2C	01	027C1	P.AFZ:	.ASCII	<1>_\
								027C3	P.AGA:	.ASCII	<1>_\	
	52	45	4C	49	41	52	54	07	027C5	P.AGB:	.ASCII	<7>\TRAILER\
							29	01	027CD	P.AGC:	.ASCII	<1>_\

BEGIN_FORMAT= P.AFO
RESET_FORMAT= P.AFQ
JOB_RESET_MODULE_FORMAT= P.AFS
INSERTION_FORMAT= P.AFU

00FC 00000 GET_QUEUE_QUALIFIERS:

CPU CODE SET_SOURCEATORS.				.WORD	Save R2,R3,R4,R5,R6,R7
57	D8	AF	9E 00002	MOVAB	INSERTION FORMAT, R7
56	0000000G	00	9E 00006	MOVAB	SYSSFAO, R6
5E		08	C2 0000D	SUBL2	#8, SP
		55	D4 00010	CLRL	INSERT FLAG
		7E	D4 00012	CLRL	CURRENT LEN
04	AE 0200	8F	3C 00014	MOVZWL	#512, STRING_PTR
	50 08	AC	D0 0001A	MOVL	STR DESC, R0
08	AE 04	AO	D0 0001E	MOVL	4(R0), STRING_PTR+4
	52 OC	AC	D0 00023	MOVL	RET LEN, R2
		62	B4 00027	CLRW	(R2)
		04	AE 9F 00029	PUSHAB	STRING PTR
		04	AE 9F 0002C	PUSHAB	CURRENT LEN
		CC	A7 9F 0002F	PUSHAB	BEGIN FORMAT
	66		03 FB 00032	CALLS	#3, SYSSFAO
	62		6E AO 00035	ADDW2	CURRENT LEN, (R2)
08	AE		6E CO 00038	ADDL2	CURRENT LEN, STRING_PTR+4
04	AE		62 3C 0003C	MOVZWL	(R2), STRING_PTR
	8F 04	AE	C3 00040	SUBL3	STRING_PTR, #512, STRING_PTR
	53 04	AC	D0 0004A	MOVL	SCB, R3
	54 0154	C3	9E 0004E	MOVAB	340(R3), R4
10	64	05	E0 00053	BBS	#5, (R4), 1\$
0C	64	04	E0 00057	BBS	#4, (R4), 1\$
	08 01	A4	E8 0005B	BLBS	1(R4), 1\$
	03 0080	C3	E8 0005F	BLBS	176(R3), 1\$
		0168	31 00064	BRW	9\$
		04	AE 9F 00067	PUSHAB	STRING_PTR
		04	AE 9F 0006A	PUSHAB	CURRENT_LEN

			F4	A7	9F	0006D	PUSHAB	JOB_RESET_MODULE_FORMAT	:	
			66	03	FB	00070	CALLS	#3, SYSSFAO	5521	
			62	6E	A0	00073	ADDW2	CURRENT_LEN, (R2)	5522	
			08	6E	C0	00076	ADDL2	CURRENT_LEN, STRING_PTR+4	5523	
04	AE 00000200	04	AE	62	3C	0007A	MOVZWL	(R2), STRING_PTR		
			26	8F	C3	0007E	SUBL3	STRING_PTR, #512, STRING_PTR		
			64	04	E1	00088	BBC	#5, (R4), 2\$	5525	
				08	A7	9F	PUSHAB	P.AFW	5526	
				08	AE	9F	PUSHAB	STRING_PTR	5527	
				08	AE	9F	PUSHAB	CURRENT_LEN	5528	
					57	DD	PUSHL	R7		
					04	FB	CALLS	#4, SYSSFAO		
					55	01	DO	#1, INSERT FLAG	5535	
					62	6E	0009A	ADDW2	CURRENT_LEN, (R2)	5536
					08	AE	A0	ADDL2	CURRENT_LEN, STRING_PTR+4	5537
04	AE 00000200	04	AE	50	AE	62	CO	MOVZWL	(R2), STRING_PTR	5538
				8F	8F	62	000A0	SUBL3	STRING_PTR, #512, STRING_PTR	
				64	64	62	3C	BBC	#4, (R4), 4\$	5541
					23	04	E1	BLBC	INSERT_FLAG, 3\$	5542
						04	000B2	PUSHAB	P.AFX	5550
						55	E9	PUSHAB	STRING_PTR	
						08	000B6	PUSHAB	CURRENT_LEN	
						08	AE	PUSHL	R7	
						57	DD	CALLS	#4, SYSSFAO	
						04	000C4	ADDW2	CURRENT_LEN, (R2)	5552
						62	6E	ADDL2	CURRENT_LEN, STRING_PTR+4	5553
04	AE 00000200	04	AE	26	AE	62	CO	MOVZWL	(R2), STRING_PTR	5554
				8F	8F	62	3C	SUBL3	STRING_PTR, #512, STRING_PTR	
				64	64	62	C3	BBC	#4, (R4), 4\$	5557
					10	A7	000DC	PUSHAB	P.AFY	5565
					08	AE	9F	PUSHAB	STRING_PTR	
					08	AE	9F	PUSHAB	CURRENT_LEN	
					57	DD	PUSHL	R7		
					04	FB	CALLS	#4, SYSSFAO		
					55	01	DO	#1, INSERT FLAG	5567	
					62	6E	A0	ADDW2	CURRENT_LEN, (R2)	5568
					08	AE	62	ADDL2	CURRENT_LEN, STRING_PTR+4	5569
04	AE 00000200	04	AE	50	AE	50	CO	MOVZWL	(R2), STRING_PTR	5570
				8F	8F	23	3C	SUBL3	STRING_PTR, #512, STRING_PTR	
				64	64	23	C3	BBC	176(R3), 6\$	5573
					04	00B0	E9	BLBC	INSERT_FLAG, 5\$	5574
					04	00B0	00106	PUSHAB	P.AFZ	5582
					15	A7	9F	PUSHAB	STRING_PTR	
					08	AE	9F	PUSHAB	CURRENT_LEN	
					08	AE	9F	PUSHL	R7	
					57	DD	CALLS	#4, SYSSFAO		
					66	62	6E	ADDW2	CURRENT_LEN, (R2)	5584
					08	AE	62	ADDL2	CURRENT_LEN, STRING_PTR+4	5585
					04	AE	62	MOVZWL	(R2), STRING_PTR	5586
04	AE 00000200	04	AE	25	AE	62	CO	SUBL3	STRING_PTR, #512, STRING_PTR	
				8F	8F	25	C3	BBC	176(R3), 6\$	5589
				64	64	00B0	E9	PUSHAB	176(R3)	5597
					04	00B0	00131	PUSHAB	STRING_PTR	
					08	AE	9F	PUSHAB	CURRENT_LEN	
					08	AE	9F	PUSHAB	RESET FORMAT	
					E0	A7	9F	CALLS	#4, SYSSFAO	
					66	62	00143	ADDW2	CURRENT_LEN, (R2)	5599
					62	6E	A0	PUSHAB		

SEPARATE
V04-001

Print Symbiont -- separation routine
GET_QUEUE_QUALIFIERS - Gets the qual

E 9
16-Sep-1984 02:23:03
14-Sep-1984 22:32:26

VAX-11 BLiss-32 V4.0-742
[PRTSMB.SRC]SEPARATE.B32;2

Page 156
(35)

SEP
V04

; Routine Size: 483 bytes, Routine Base: CODE + 27CF

4750
4751
4752
4753
4754
4755
4756
4757
4758
4759
4760
4761
4762
4763
4764
4765
4766
4767
4768
4769
4770
4771
4772
4773
4774
4775
4776
4777
4778
4779
4780
4781
4782
4783
4784
4785
4786
4787
4788
4789
4790
4791
4792
4793
4794
4795
4796
4797
4798
4799
4800
4801
4802
4803
4804
4805
4806

5662 1 %sbttl 'GET_FORM_QUALIFIERS - Get the qualifiers which pertain to forms'
5663 1 !++
5664 1 Functional Description:
5665 1 This routine returns a string containing the all relevant file qualifier
5666 1 information.
5667 1 Formal Parameters:
5668 1 SCB - Address of the SCB
5669 1 STR_DESC - Desc of String to Return
5670 1 RET_LEN - Return length of Desc.
5671 1
5672 1 Implicit Inputs:
5673 1 none
5674 1
5675 1 Implicit Outputs:
5676 1 none
5677 1
5678 1 Returned Value:
5679 1 none
5680 1
5681 1 Side Effects:
5682 1 none
5683 1
5684 1 --
5685 1 ROUTINE GET_FORM_QUALIFIERS {
5686 1 SCB : REF \$BLOCK,
5687 1 STR_DESC : REF VECTOR[?],
5688 1 RET_LEN : REF VECTOR [,WORD] , : NOVALUE = ! SCB
5689 1 ! Output buffer desc
5690 2 BEGIN ! Return length (word)
5691 2 BIND
P 5692 2 BEGIN FORMAT = \$DESCRIPTOR(
5693 2 'Form Qualifiers:');
5694 2 LITERAL
5695 2 K_MAX_BUFFER_SIZE = 512;
5696 2
5697 2 LOCAL
5698 2 CURRENT_LEN : INITIAL (0),
5699 2 STRING_PTR : VECTOR [2]; ! Pointer to current string
5700 2
5701 2
5702 2
5703 2 STRING_PTR[SIZE] = K_MAX_BUFFER_SIZE;
5704 2 STRING_PTR[ADDR] = .STR_DESC[ADDR]; ! init address
5705 2
5706 2 RET_LEN[0] = 0;
5707 2
P 5708 2 \$FAO (BEGIN FORMAT,
P 5709 2 CURRENT_LEN,
P 5710 2 STRING_PTR[0], ! return length
5711 2); ! address of string
5712 2
5713 2 RET_LEN[0] = .RET_LEN[0] + .CURRENT_LEN;
5714 2 STRING_PTR[ADDR] = .STRING_PTR[ADDR] + .CURRENT_LEN;
5715 2 STRING_PTR[SIZE] = K_MAX_BUFFER_SIZE - .RET_LEN[0];
5716 2
5717 2
5718 2

```

: 4807      5719 2 : Don't print anything if no flags were set
: 4808      5720 2
: 4809      5721 2 IF .RET_LEN[0] LEQ 18
: 4810      5722 2 THEN
: 4811      5723 2     RET_LEN[0] = 0;
: 4812      5724 2
: 4813      5725 2 ! Length returned must be less than max string size
: 4814      5726 2 IF .RET_LEN[0] GTR K_MAX_BUFFER_SIZE
: 4815      5727 2 THEN
: 4816      5728 3 BEGIN
: 4817      5729 3     RET_LEN[0] = 512;
: 4818      5730 3     RETURN;
: 4819      5731 2 END;
: 4820      5732 2
: 4821      5733 1 END;

```

73 72 65 69 66 69 6C 61 75 51 20 6D 72 6F 46 029B2 P.AGE: .ASCII \Form Qualifiers\
 3A 029C1
 029C2
 00000010 029C4 P.AGD: .BLKB 2
 00000000 029C8 .LONG 16
 .ADDRESS P.AGE

BEGIN_FORMAT= P.AGD

0004 00000 GET_FORM_QUALIFIERS:					
				.WORD	Save R2
	SE		08 C2 00002	SUBL2	#8, SP
04	AE	0200	7E D4 00005	CLRL	CURRENT LEN
	50	08	3C 00007	MOVZWL	#512, STRING_PTR
08	AE	04	AC D0 0000D	MOVL	STR_DESC, R0
	52	OC	D0 00011	MOVL	4(R0), STRING_PTR+4
			62 B4 0001A	MOVL	RET LEN, R2
			04 AE 9F 0001C	CLRW	(R2)
			04 AE 9F 0001F	PUSHAB	STRING_PTR
			D3 AF 9F 00022	PUSHAB	CURRENT LEN
		00000000G	00	PUSHAB	BEGIN FORMAT
			03 FB 00025	CALLS	#3, \$PSSFAO
			62 A0 0002C	ADDW2	CURRENT LEN, (R2)
08	AE	0200	6E C0 0002F	ADDL2	CURRENT LEN, STRING_PTR+4
	04	AE	62 3C 00033	MOVZWL	(R2), STRING_PTR
04	AE	00000200	8F 04	SUBL3	STRING_PTR, #512, STRING_PTR
			12	CMPW	(R2), #18
			02 1A 00041	BGTRU	1\$
			02 B1 00044	CLRW	(R2)
			62 B4 00046	CMPW	(R2), #512
			62 B1 00048	BLEQU	2\$
			05 1B 0004D	MOVW	#512, (R2)
			62 0200 8F	RET	
			04 00054 2\$:		

; Routine Size: 85 bytes, Routine Base: CODE + 29CC

```

: 4823      5734 1 %sbttl 'GET_USER_NOTE- Insert a Note into the Page'
: 4824      5735 1 ++
: 4825      5736 1 Functional Description:
: 4826      5737 1 This routine gets a note as specified by the user for the frame.
: 4827      5738 1
: 4828      5739 1 Formal Parameters:
: 4829      5740 1 SCB          - Address of the SCB
: 4830      5741 1 STR_DESC     - Desc of String to Return
: 4831      5742 1 RET_LEN      - Return length of Desc.
: 4832      5743 1
: 4833      5744 1 Implicit Inputs:
: 4834      5745 1             none
: 4835      5746 1
: 4836      5747 1 Implicit Outputs:
: 4837      5748 1             none
: 4838      5749 1
: 4839      5750 1 Returned Value:
: 4840      5751 1             none
: 4841      5752 1
: 4842      5753 1 Side Effects:
: 4843      5754 1             none
: 4844      5755 1 --
: 4845      5756 1 ROUTINE GET_USER_NOTE (
: 4846      5757 1           SCB          : REF SBBLOCK,          ! SCB
: 4847      5758 1           STR_DESC     : REF VECTOR[2],       ! Output buffer desc
: 4848      5759 1           RET_LEN      : REF VECTOR [,WORD]   ! Return length (word)
: 4849      5760 1           )           : NOVALUE =
: 4850      5761 2 BEGIN
: 4851      5762 2 BIND
: P 5763 2 NOTE_FULL_FORMAT = $DESCRIPTOR(
: 4853      5764 2     "NOTE: !AF"           );           ! - user note
: 4854      5765 2 LITERAL
: 4855      5766 2     K_MIN_NOTE_LEN = 6;
: 4856      5767 2 LOCAL
: 4857      5768 2     STATUS;
: 4858      5769 2
: P 5770 2 STATUS = $FAO ( NOTE_FULL_FORMAT,
: 4859      5771 2           RET[ENCO],
: 4860      5772 2           STR_DESC[0],
: 4861      5773 2           .SCB_SIZE_(NOTE),
: 4862      5774 2           .SCB_ADDR_(NOTE));    ! user note
: 4863      5775 2
: 4864      5776 2 IF .RET_LEN[0] LEQ K_MIN_NOTE_LEN        ! print nothing... no note
: 4865      5777 2     THEN RET_LEN[0] = 0;
: 4866      5778 2
: 4867      5779 2 RETURN SSS_NORMAL;
: 4868      5780 2
: 4869      5781 1 END;

```

```

46 41 21 20 3A 45 54 4F 4E 02A21 P.AGG: .ASCII \NOTE: !AF\
02A2A               .BLKB 2
00000009 02A2C P.AGF: .LONG 9
00000000 02A30     .ADDRESS P.AGG

```

NOTE_FULL_FORMAT= P.AGF

0000 00000 GET_USER_NOTE:
50 04 AC DD 00002 .WORD Save nothing : 5756
00D8 CO DD 00006 MOVL SCB, R0 : 5774
7E 00D4 CO 3C 0000A PUSHL 216(R0)
08 AC DD 0000F MOVZWL 212(R0), -(SP)
OC AC DD 00012 PUSHL STR DESC
EO AF 9F 00015 PUSHAB RET-LEN
00000000G 00 05 FB 00018 CALLS NOTE FULL FORMAT
06 OC BC B1 0001F CMPW #5, SYSSFAO
03 1A 00023 BGTRU @RET_LEN, #6 : 5776
OC BC B4 00025 CLRW @RET_LEN : 5777
04 00028 1\$: RET : 5781

; Routine Size: 41 bytes, Routine Base: CODE + 2A34

```

4872      5782 1 %sbttl 'GET_RECEIPT_BOX - Insert a "Received Box" into the Page'
4873      5783 1 ++
4874      5784 1 Functional Description:
4875      5785 1 This routine gets a note as specified by the user for the frame.
4876      5786 1
4877      5787 1 Formal Parameters:
4878      5788 1 SCB          - Address of the SCB
4879      5789 1 STR_DESC     - Desc of String to Return
4880      5790 1 RET_LEN      - Return length of Desc.
4881      5791 1
4882      5792 1 Implicit Inputs:
4883      5793 1 none
4884      5794 1
4885      5795 1 Implicit Outputs:
4886      5796 1 none
4887      5797 1
4888      5798 1 Returned Value:
4889      5799 1 none
4890      5800 1
4891      5801 1 Side Effects:
4892      5802 1 none
4893      5803 1 --
4894      5804 1 ROUTINE GET_RECEIPT_BOX (
4895      5805 1           SCB          : REF $BLOCK,
4896      5806 1           STR_DESC     : REF VECTOR[2],
4897      5807 1           RET_LEN      : REF VECTOR [,WORD]
4898      5808 1           )          : NOVALUE =
4899      5809 2 BEGIN
4900      5810 2 BIND
4901      P 5811 2 NOTE132_FORMAT = $DESCRIPTOR (
4902      P 5812 2
4903      P 5813 2
4904      P 5814 2
4905      P 5815 2
4906      P 5816 2
4907      P 5817 2
4908      P 5818 2
4909      P 5819 2
4910      5820 2 ) : VECTOR; ! - receipt box
4911      5821 2
4912      5822 2 CH$MOVE (.NOTE132_FORMAT[0], .NOTE132_FORMAT[1], .STR_DESC[ADDR]);
4913      5823 2 RET_LEN[0] = .NOTE132_FORMAT[0];
4914      5824 2
4915      5825 2 RETURN SSS_NORMAL;
4916      5826 1 END;

```

2D 2B 02A5D P.AGI:	.ASCII \+-----+\\	:
2D 02A6C	.ASCII \\	:\\
20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 21 02A7B	.ASCII \\	:\\
20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 02A85	.ASCII \\	:\\
2E 2E 2E 20 3A 64 65 76 69 65 63 65 52 20 21 02AA3	.ASCII \\ Received: !\\	:\\
2E 2E 2E 2E 2E 2E 21 20 2E 2E 2E 2E 2E 2E 2E 2E 2E 02ABC	.ASCII \\ Received: !\\	:\\

SEPARATE
V04-001 Print Symbiont -- separation routines
GET_RECEIPT_BOX - Insert a "Received Box" into

K 9
16-Sep-1984 02:23:03
14-Sep-1984 22:32:26

VAX-11 Bliss-32 V4.0-742
[PRTSMB.SRC]SEPARATE.B32;2

Page 162
(38)

```

20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 02AD5 .ASCII \!
20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 02AE4 .\.
2E 2E 2E 20 3A 20 20 20 20 20 20 20 20 20 20 20 20 20 20 02AF3 .\.
2E 2E 2E 2E 2E 21 20 20 20 20 20 20 20 20 20 20 20 20 20 02B0C .\.
20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 02B1B .\.
20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 02B25 .\.
20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 02B34 .\.
2E 2E 2E 20 3A 72 6F 74 61 72 65 70 4F 21 02B4D .\.
2E 2E 2E 2E 2E 21 20 20 20 20 20 20 20 20 20 20 20 02B5C .\.
20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 02B68 .\.
20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 02B75 .\.
20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 02B84 .\.
20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 02B93 .\.
20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 02B9D .\.
00000140 02BA0 P.AGH: .BLKB 3 .\.
00000000 02BA4 .ADDRESS P.AGI .\.

NOTE132_FORMAT= P.AGH

```

003C 00000 GET_RECEIPT_BOX:

04 80	F1 0C	50 BF	08 E8	AC AF	D0 28	00002 00006	.WORD	Save R2,R3,R4,R5	5804
							MOVL	STR DESC, R0	5822
							MOV3	NOTE132_FORMAT, @NOTE132_FORMAT+4, @RET_LEN	5823
							MOVW	NOTE132_FORMAT, @RET_LEN	5826
							RET		

; Routine Size: 19 bytes, Routine Base: CODE + 2BAB

```

5827 1 Xsbttl 'GET_RULER_FINE - Insert a fine "RULER" into the Page'
5828 1 ++
5829 1 Functional Description:
5830 1 This routine gets a fine ruler '1234567890'
5831 1 Formal Parameters:
5832 1 SCB      - Address of the SCB
5833 1 STR_DESC - Desc of String to Return
5834 1 RET_LEN  - Return length of Desc.
5835 1
5836 1 Implicit Inputs:
5837 1 none
5838 1
5839 1 Implicit Outputs:
5840 1 none
5841 1
5842 1 Returned Value:
5843 1 none
5844 1
5845 1 Side Effects:
5846 1 none
5847 1 --
5848 1 ROUTINE GET_RULER_FINE (
5849 1 SCB      : REF $BLOCK,
5850 1 STR_DESC : REF VECTOR[2],
5851 1 RET_LEN  : REF VECTOR [,WORD]
5852 1           ) : NOVALUE =
5853 1           ! SCB
5854 2           ! Output buffer desc
5855 2           ! Return length (word)
5856 2 BEGIN
5857 2 BIND
5858 2 VMS_FORMAT = $DESCRIPTOR (
5859 2     '1234567890'
5860 2 );
5861 2 $FAO ( VMS_FORMAT,
5862 2     RET_LEN[0],
5863 2     STR_DESC[0]
5864 2 );
5865 2 RETURN SSS_NORMAL;
5866 1 END;

```

30	39	38	37	36	35	34	33	32	31	02BBB	P.AGK:	.ASCII	\1234567890\	
										02BC5		.BLKB	3	
										0000000A	02BC8	P.AGJ:	.LONG	10
										00000000	02BCC		.ADDRESS	P.AGK

VMS_FORMAT= P.AGJ

0000 00000 GET_RULER_FINE:											
08	AC	DD	00002	QORD	Save nothing						
0C	AC	DD	00005	PUSHL	STR_DESC						
ED	AF	9F	00008	PUSHL	RET_LEN						
0000000G 00				PUSHAB	VMS_FORMAT						
				CALLS	#3,-SYSSFAO						

: 5849

: 5863

SEPARATE
V04-001

Print Symbiont -- separation routines
GET_RULER_FINE - Insert a fine "RULER" into the

M 9

16-Sep-1984 02:23:03

14-Sep-1984 22:32:26

VAX-11 Bliss-32 V4.0-742
[PRTSMB.SRC]SEPARATE.B32;2

Page 164
(39)

; 5866

04 00012 RET

; Routine Size: 19 bytes. Routine Base: CODE + 28D0

```

: 4959      5867 1 %sbttl 'GET_RULER_COARSE - Insert a coarse "RULER" into the Page'
: 4960      5868 1 ++
: 4961      5869 1 Functional Description:
: 4962      5870 1 This routine gets a coarse ruler '1...2...3...' for the frame.
: 4963      5871 1
: 4964      5872 1 Formal Parameters:
: 4965      5873 1      SCB          - Address of the SCB
: 4966      5874 1      STR_DESC    - Desc of String to Return
: 4967      5875 1      RET_LEN     - Return Length of Desc.
: 4968      5876 1
: 4969      5877 1 Implicit Inputs:
: 4970      5878 1      none
: 4971      5879 1
: 4972      5880 1 Implicit Outputs:
: 4973      5881 1      none
: 4974      5882 1
: 4975      5883 1 Returned Value:
: 4976      5884 1      none
: 4977      5885 1
: 4978      5886 1 Side Effects:
: 4979      5887 1      none
: 4980      5888 1 --
: 4981      5889 1 ROUTINE GET_RULER_COARSE (
: 4982      5890 1      SCB          : REF $BLOCK
: 4983      5891 1      STR_DESC    : REF VECTOR[2]
: 4984      5892 1      RET_LEN     : REF VECTOR [,WORD]   ! SCB
: 4985      5893 1              ) : NOVALUE =           ! Output buffer desc
: 4986      5894 2 BEGIN
: 4987      5895 2 BIND
: 4988      P 5896 2 VMS_FORMAT = $DESCRIPTOR (
: 4989      P 5897 2      '1'
: 4990      P 5898 2      '2'
: 4991      P 5899 2      '3'
: 4992      P 5900 2      '4'
: 4993      P 5901 2      '5'
: 4994      P 5902 2      '6'
: 4995      P 5903 2      '7'
: 4996      P 5904 2      '8'
: 4997      P 5905 2      '9'
: 4998      P 5906 2      '0'
: 4999      P 5907 2      );
: 5000      P 5908 2
: 5001      P 5909 2 SFAO ( VMS_FORMAT,
: 5002      P 5910 2      RET_LEN[0]
: 5003      P 5911 2      STR_DESC[0]
: 5004      P 5912 2      );
: 5005      P 5913 2
: 5006      P 5914 2 RETURN $SS_NORMAL;
: 5007      P 5915 1 END;

```

31	20	20	20	20	20	20	20	20	20	20	02BE3	P.AGM:	.ASCII	\	1\
32	20	20	20	20	20	20	20	20	20	20	02BED		.ASCII	/	2\
33	20	20	20	20	20	20	20	20	20	20	02BF7		.ASCII	/	3\
34	20	20	20	20	20	20	20	20	20	20	02C01		.ASCII	/	4\
35	20	20	20	20	20	20	20	20	20	20	02C0B		.ASCII	/	5\

SEPARATE
V04-001Print Symbiont -- separation routines
GET_RULER_COARSE - Insert a coarse "RULER" intoB 10
16-Sep-1984 02:23:03
14-Sep-1984 22:32:26VAX-11 BLISS-32 V4.0-742
[PRTSMB.SRC]SEPARATE.B32;2Page 166
(40)

36	20	20	20	20	20	20	20	20	20	20	02C15	.ASCII	'	6\
37	20	20	20	20	20	20	20	20	20	20	02C1F	.ASCII	'	7\
38	20	20	20	20	20	20	20	20	20	20	02C29	.ASCII	'	8\
39	20	20	20	20	20	20	20	20	20	20	02C33	.ASCII	'	9\
30	20	20	20	20	20	20	20	20	20	20	02C3D	.ASCII	'	0\
											02C47	BLKB	1	
											00000064,	02C48	P.AGL:	.LONG 100
											00000000,	02C4C		.ADDRESS P.AGM

VMS_FORMAT= P.AGL

0000 0000 GET_RULER_COARSE:														
08	AC	DD	00002	.WORD	Save nothing									5889
0C	AC	DD	00005	PUSHL	STR_DESC									5912
ED	AF	9F	00008	PUSHL	RET_LEN									
00000000G 00	03	FB	0000B	PUSHAB	VMS_FORMAT									
	04	00012	CALLS	#3, SYSSFAO										
			RET											5915

: Routine Size: 19 bytes. Routine Base: CODE + 2C50

SEF
V04

```

: 5009      5916 1 %sbttl 'GET_FORM_SIZE - Determine the Size of Form Specified'
: 5010      5917 1 ++
: 5011      5918 1 Functional Description:
: 5012      5919 1 This routine determines the which standard form is
: 5013      5920 1 specified by interrogating the SCB for length and width
: 5014      5921 1 Standard forms sizes include: 132x66, 132x51, 80x66, 80x51 or
: 5015      5922 1 40xany_length. Otherwise form size is 'non_std'.
: 5016      5923 1
: 5017      5924 1 Formal Parameters:
: 5018      5925 1          SCB      - Address of the SCB
: 5019      5926 1
: 5020      5927 1 Implicit Inputs:
: 5021      5928 1          none
: 5022      5929 1
: 5023      5930 1 Implicit Outputs:
: 5024      5931 1          none
: 5025      5932 1
: 5026      5933 1 Returned Value:
: 5027      5934 1          none
: 5028      5935 1
: 5029      5936 1 Side Effects:
: 5030      5937 1          none
: 5031      5938 1 --
: 5032      5939 1 ROUTINE GET_FORM_SIZE (
: 5033      5940 1          SCB      : REF $BBLOCK
: 5034      5941 1          ): NOVALUE =
: 5035      5942 2 BEGIN
: 5036      5943 2
: 5037      5944 2          SCB[PSMSL_PAGE_WIDTH] = .SCB[PSMSL_FORM_WIDTH];
: 5038      5945 2          IF .SCB[PSMSL_PAGE_WIDTH] GTRU 200
: 5039      5946 2          THEN
: 5040      5947 2          SCB[PSMSL_PAGE_WIDTH] = 200;
: 5041      5948 2
: 5042      5949 2          SCB[PSMSL_PAGE_LENGTH] = .SCB[PSMSL_FORM_LENGTH];
: 5043      5950 2          IF .SCB[PSMSL_PAGE_LENGTH] GTRU 100
: 5044      5951 2          THEN
: 5045      5952 2          SCB[PSMSL_PAGE_LENGTH] = 100;
: 5046      5953 2
: 5047      5954 2          WHILE .SCB[PSMSL_PAGE_LENGTH] LSSU 40
: 5048      5955 2          DO
: 5049      5956 2          SCB[PSMSL_PAGE_LENGTH] =
: 5050      5957 2          .SCB[PSMSL_PAGE_LENGTH] + .SCB[PSMSL_FORM_LENGTH];
: 5051      5958 2
: 5052      5959 1 END;

```

0000 00000 GET_FORM_SIZE:						
					WORD	Save nothing
	50	04	AC	D0	00002	MOVL SCB, R0
00000008	51	0200	C0	9E	00006	MOVAB 512(R0), R1
	61	008C	C0	D0	00008	MOVL 140(R0), (R1)
			61	D1	00010	CMPL (R1), #200
				04	1B	BLEQU 1\$
				8F	9A	MOVZBL #200, (R1)
	61	C8				

: 5939
: 5944
: 5945
: 5947

SEPARATE
V04-001

Print Symbiont -- separation routines
GET_FORM_SIZE - Determine the Size of Form Spec

D 10

16-Sep-1984 02:23:03
14-Sep-1984 22:32:26

VAX-11 Bliss-32 V4.0-742
[PRTSMB.SRC]SEPARATE.B32;2

Page 168
(41)

00000064	51	01F8	C0	9E	0001D	1\$:	MOVAB	504(R0), R1
	61	78	A0	D0	00022		MOVL	120(R0), (R1)
	8F		61	D1	00026		CMPL	(R1), #100
			04	1B	0002D		BLEQU	2\$
	28	64	8F	9A	0002F		MOVZBL	#100, (R1)
			61	D1	00033	2\$:	CMPL	(R1), #40
			06	1E	00036		BGEQU	3\$
	61	78	A0	C0	00038		ADDL2	120(R0), (R1)
			F5	11	0003C		BRB	2\$
			04	0003E	3\$:		RET	

; Routine Size: 63 bytes, Routine Base: CODE + 2063

```
: 5054      5960 1 %sbttl 'FILL_FRAME - Insert Information into this Frame of the Page'
: 5055      5961 1 ++
: 5056      5962 1 Functional Description:
: 5057      5963 1 This procedure inserts a character into an array(frame)
: 5058      5964 1 until no room left in the frame.
: 5059      5965 1
: 5060      5966 1 Formal Parameters:
: 5061      5967 1 SCB           - Address of the SCB
: 5062      5968 1 CHAR          - Descriptor of String to Insert
: 5063      5969 1 FRAME_PTR     - Address of first byte of Frame
: 5064      5970 1 FRAME_LENGTH   - Length of Frame
: 5065      5971 1 FRAME_WIDTH    - Width of Frame
: 5066      5972 1
: 5067      5973 1 Implicit Inputs:
: 5068      5974 1             none
: 5069      5975 1
: 5070      5976 1 Implicit Outputs:
: 5071      5977 1             none
: 5072      5978 1
: 5073      5979 1 Returned Value:
: 5074      5980 1             none
: 5075      5981 1
: 5076      5982 1 Side Effects:
: 5077      5983 1             none
: 5078      5984 1 --
: 5079      5985 1 ROUTINE FILL_FRAME (
: 5080      5986 1             SCB           : REF $BBLOCK,
: 5081      5987 1             CHAR          :
: 5082      5988 1             FRAME_PTR     : REF PAGE_ARRAY,
: 5083      5989 1             FRAME_WIDTH   :
: 5084      5990 1             FRAME_LENGTH  :
: 5085      5991 1             ): NOVALUE =
: 5086      5992 2 BEGIN
: 5087      5993 2 LOCAL  PTR           : REF PAGE_ARRAY,
: 5088      5994 2             LOC_FRAME_LENGTH,
: 5089      5995 2             LOC_FRAME_WIDTH ;
: 5090      5996 2
: 5091      5997 2
: 5092      5998 2 ! Check for dumb calls
: 5093      5999 2
: 5094      6000 2 IF (.FRAME_LENGTH LEQ 0) OR (.FRAME_WIDTH LEQ 0)
: 5095      6001 2 THEN
: 5096      6002 2     RETURN;
: 5097      6003 2 IF (.FRAME_LENGTH GTR .SCB[PSMSL_PAGE_LENGTH]) OR
: 5098      6004 2             (.FRAME_WIDTH GTR .SCB[PSMSL_PAGE_WIDTH])
: 5099      6005 2 THEN
: 5100      6006 2     RETURN;
: 5101      6007 2
: 5102      6008 2 ! Check page boundary conditions
: 5103      6009 2
: 5104      6010 2 LOC_FRAME_LENGTH = .FRAME_LENGTH;
: 5105      6011 2 IF .LOC_FRAME_LENGTH GTR .SCB[PSMSL_PAGE_LENGTH]
: 5106      6012 2 THEN
: 5107      6013 2     LOC_FRAME_LENGTH = .SCB[PSMSL_PAGE_LENGTH];
: 5108      6014 2
: 5109      6015 2 LOC_FRAME_WIDTH = .FRAME_WIDTH;
: 5110      6016 2 IF .LOC_FRAME_WIDTH GTR .SCB[PSMSL_PAGE_WIDTH]
```

! stay in page bounds

```

: 5111    6017 2 THEN
: 5112    6018 2      LOC_FRAME_WIDTH = .SCB[PSMSL_PAGE_WIDTH];
: 5113    6019 2
: 5114    6020 2
: 5115    6021 2      PTR = FRAME_PTR[0,0,.SCB[PSMSL_PAGE_WIDTH]];
: 5116    6022 2
: 5117    6023 2
: 5118    6024 2      DECR L FROM .LOC_FRAME_LENGTH TO 1 DO
: 5119    6025 3      BEGIN
: 5120    6026 3          CH$FILL( .CHAR, .LOC_FRAME_WIDTH,,PTR);
: 5121    6027 3
: 5122    6028 3      PTR = .PTR + .SCB[PSMSL_PAGE_WIDTH];
: 5123    6029 3
: 5124    6030 2      END;
: 5125    6031 1 END;

```

! stay in page bounds

! Address calc. is based
on Form Width

03FC 00000 FILL_FRAME:

					.WORD	Save R2,R3,R4,R5,R6,R7,R8,R9		5985
	52	14	AC D0 00002		MOVL	FRAME_LENGTH, R2		6000
		10	4C 15 00006		BLEQ	\$		
			AC D5 00008		TSTL	FRAME_WIDTH		
	50	04	47 15 0000B		BLEQ	\$		
	51	01F8	AC D0 0000D		MOVL	SCB, R0		6003
	51		C0 D0 00011		MOVL	504(R0), R1		
			52 D1 00016		CMPL	R2, R1		
			39 14 00019		BGTR	\$		
	0200	C0	AC D1 0001B		CMPL	FRAME_WIDTH, 512(R0)		6004
			31 14 00021		BGTR	\$		
	56		52 D0 00023		MOVL	R2, LOC_FRAME_LENGTH		6010
	51		56 D1 00026		CMPL	LOC_FRAME_LENGTH, R1		6011
			03 15 00029		BLEQ	1\$		
	56		51 D0 00028		MOVL	R1, LOC FRAME LENGTH		6013
	59	10	AC D0 0002E	1\$:	MOVL	FRAME WIDTH, LOC_FRAME_WIDTH		6015
	58	0200	C0 D0 00032		MOVL	512(R0), R8		6016
	58		59 D1 00037		CMPL	LOC_FRAME_WIDTH, R8		
			03 15 0003A		BLEQ	2\$		
	59		58 D0 0003C		MOVL	R8, LOC FRAME WIDTH		6018
	57	0C	AC D0 0003F	2\$:	MOVL	FRAME_PTR, PTR		6021
			56 D6 00043		INCL	L		6026
			0A 11 00045		BRB	4\$		
59	08	AC	6E	00 2C 00047	3\$:	MOVCS	#0, (SP), CHAR, LOC_FRAME_WIDTH, (PTR)	
				67 0004D				
	57		58 C0 0004E		ADDL2	R8, PTR		6028
	F3		56 F5 00051	4\$:	SOBGTR	L, 3\$		6024
			04 00054	5\$:	RET			6031

; Routine Size: 85 bytes, Routine Base: CODE + 2CA2

```
: 5127    6032 1 %sbttl 'SCROLL_FRAME - Insert Information into this Frame of the Page'
: 5128    6033 1 ++
: 5129    6034 1 : Functional Description:
: 5130    6035 1 : This procedure inserts a string into an array(frame) repeatedly
: 5131    6036 1 : until no room is left in the frame.
: 5132    6037 1 :
: 5133    6038 1 : Formal Parameters:
: 5134    6039 1 : SCB           - Address of the SCB
: 5135    6040 1 : CHAR_STRING   - Descriptor of String to Insert
: 5136    6041 1 : FRAME_PTR     - Address of first byte of Frame
: 5137    6042 1 : FRAME_LENGTH   - Length of Frame
: 5138    6043 1 : FRAME_WIDTH    - Width of Frame
: 5139    6044 1 :
: 5140    6045 1 : Implicit Inputs:
: 5141    6046 1 :             none
: 5142    6047 1 :
: 5143    6048 1 : Implicit Outputs:
: 5144    6049 1 :             none
: 5145    6050 1 :
: 5146    6051 1 : Returned Value:
: 5147    6052 1 :             none
: 5148    6053 1 :
: 5149    6054 1 : Side Effects:
: 5150    6055 1 :             none
: 5151    6056 1 :-- ROUTINE SCROLL_FRAME (
: 5152    6057 1 :             SCB           : REF $BBLOCK,
: 5153    6058 1 :             CHAR_STRING   : REF VECTOR[2],
: 5154    6059 1 :             FRAME_PTR     : REF PAGE_ARRAY,
: 5155    6060 1 :             FRAME_WIDTH    :
: 5156    6061 1 :             FRAME_LENGTH   :
: 5157    6062 1 :             ) : NOVALUE =
: 5158    6063 1 :
: 5159    6064 2 BEGIN
: 5160    6065 2 LOCAL  PTR           : REF PAGE_ARRAY,
: 5161    6066 2 :             LOC_FRAME_LENGTH,
: 5162    6067 2 :             LOC_FRAME_WIDTH ,
: 5163    6068 2 :             TEMP_PTR
: 5164    6069 2 :             START_CNT    : INITIAL (0),
: 5165    6070 2 :             CHARS
: 5166    6071 2 :             NUM_CHARS    : INITIAL (0);
: 5167    6072 2 :
: 5168    6073 2 : Check for dumb calls
: 5169    6074 2 :
: 5170    6075 2 IF (.FRAME_LENGTH LEQ 0) OR (.FRAME_WIDTH LEQ 0) OR
: 5171    6076 2 :             (.FRAME_LENGTH GTR .SCB[PSMSL_PAGE_LENGTH]) OR
: 5172    6077 2 :             (.FRAME_WIDTH GTR .SCB[PSMSL_PAGE_WIDTH])
: 5173    6078 2 THEN
: 5174    6079 2     RETURN;
: 5175    6080 2 :
: 5176    6081 2 LOC_FRAME_LENGTH = .FRAME_LENGTH;
: 5177    6082 2 LOC_FRAME_WIDTH = .FRAME_WIDTH;
: 5178    6083 2 :
: 5179    6084 2 PTR = FRAME_PTR[0,0,.SCB[PSMSL_PAGE_WIDTH]];
: 5180    6085 2 TEMP_PTR = [CHAR STRING[ADDR];
: 5181    6086 2 :
: 5182    6087 2 CHAR5 = CH$PTR(TEMP_PTR);
: 5183    6088 2 :
```

```

5184 6089 2      INCR L FROM 0 TO (.LOC_FRAME_LENGTH-1) DO
5185 6090 2      BEGIN
5186 6091 3      PTR = FRAME_PTR[0,0,.SCB[PSMSL_PAGE_WIDTH]]      ! Address calc. is based
5187 6092 3      + (.L * .SCB[PSMSL_PAGE_WIDTH]);          ! on Form Width
5188 6093 3
5189 6094 3
5190 6095 3      Move the rest of the string into the beginning of the next frame
5191 6096 3
5192 6097 4      IF (.NUM_CHARS LEQ .CHAR_STRING[SIZE])
5193 6098 4      AND T(.CHAR_STRING[SIZE]-.NUM_CHARS) LSS .LOC_FRAME_WIDTH)
5194 6099 3      THEN      ! Scroll it
5195 6100 4      BEGIN
5196 6101 4      TEMP_PTR = .CHAR_STRING[ADDR] + .NUM_CHARS; ! move remainder of str.
5197 6102 4      CH$MOVE(.CHAR_STRING[SIZE]-.NUM_CHARS, .TEMP_PTR, .PTR);
5198 6103 4      PTR = .PTR + T(.CHAR_STRING[SIZE]-.NUM_CHARS); ! incr by no. inserted
5199 6104 4
5200 6105 4      START_CNT = .CHAR_STRING[SIZE] - .NUM_CHARS;
5201 6106 4      TEMP_PTR = .CHAR_STRING[ADDR];
5202 6107 3      END;
5203 6108 3
5204 6109 3      INCR I FROM .START_CNT TO .LOC_FRAME_WIDTH BY .CHAR_STRING[SIZE] DO
5205 6110 4      BEGIN
5206 6111 5      IF .CHAR_STRING[SIZE] GEQ (.LOC_FRAME_WIDTH - .I)
5207 6112 4      THEN      NUM_CHARS = (.LOC_FRAME_WIDTH - .I)
5208 6113 5
5209 6114 4      ELSE      NUM_CHARS = .CHAR_STRING[SIZE];
5210 6115 4
5211 6116 4
5212 6117 4      CH$MOVE(.NUM_CHARS, .TEMP_PTR, .PTR);
5213 6118 4
5214 6119 4      PTR = .PTR + .NUM_CHARS;
5215 6120 3      END;
5216 6121 2      END;
5217 6122 1 END;

```

OFFC 00000 SCROLL_FRAME:						
5E		OC C2 00002		.WORD	Save R2,R3,R4,R5,R6,R7,R8,R9,R10,R11	6057
		59 D4 00005		SUBL2	#12, SP	6064
51	14	7E D4 00007		CLRL	START_CNT	
		AC D0 00009		CLRL	NUM_CHARS	6076
		03 15 0000D		MOVL	FRAME_LENGTH, R1	
	10	AC D5 0000F		BLEQ	1\$	
		01 14 00012 1\$:		TSTL	FRAME_WIDTH	
		04 00014		BGTR	2\$	
				RET		
01FB	50	04 AC D0 00015 2\$:		MOVL	SCB, R0	6077
	C0	51 D1 00019		CMPL	R1, 504(R0)	
		0A 14 0001E		BGTR	3\$	
0200	50	04 AC D0 00020		MOVL	SCB, R0	6078
	C0	01 D1 00024		CMPL	FRAME_WIDTH, 512(R0)	
		01 15 0002A 3\$:		BLEQ	4\$	
		04 0002C		RET		
OC	AE	51 D0 0002D 4\$:		MOVL	R1, LOC_FRAME_LENGTH	6082

SEPARATE
V04-001

Print Symbiont -- separation routines
SCROLL_FRAME - Insert Information into

13 10
65

16-Sep-1984 02:23:03
14-Sep-1984 22:32:26

VAX-11 Bliss-32 V4.0-742
[PRTSMB.SRC]SEPARATE.B32;2

Page 173
(43)

SE
VO

		58	10	AC	DO	00031		MOVL	FRAME_WIDTH, LOC_FRAME_WIDTH
	04	AE	0C	AC	DO	00035		MOVL	FRAME_PTR, PTR
		5A	08	AC	DO	0003A		MOVL	CHAR_STRING, R10
		5B	04	AA	DO	0003E		MOVL	4(R10), TEMP_PTR
		50	5B	SB	DO	00042		MOVL	TEMP_PTR, CHARS
	08	AE	04	AC	DO	00045		MOVL	SCB, -8(SP)
		57		01	CE	0004A		MNEGL	#1, L
				61	11	0004D		BRB	11\$
51	08	AE	00000200	8F	C1	0004F	5\$:	ADDL3	#512, 8(SP), R1
50		57		61	C5	00058		MULL3	(R1), L, R0
	04	AE	0C BC40	9E	0005C			MOVAB	FRAME_PTR[R0], PTR
		6A		6E	D1	00062		CMPL	NUM_CHARS, (R10)
				22	14	00065		BGTR	6\$
56		6A		6E	C3	00067		SUBL3	NUM_CHARS, (R10), R6
		58		56	D1	0006B		CMPL	R6, LOC_FRAME_WIDTH
				19	18	0006E		BGEQ	6\$
5B	04	AA		6E	C1	00070		ADDL3	NUM_CHARS, 4(R10), TEMP_PTR
56		6A		6E	C3	00075		SUBL3	NUM_CHARS, (R10), R6
04	BE	6B		56	28	00079		MOVC3	R6, (TEMP_PTR), APTR
		04	AE	56	CO	0007E		ADDL2	R6, PTR
		59		56	DO	00082		MOVL	R6, START_CNT
		58	04	AA	DO	00085		MOVL	4(R10), TEMP_PTR
		56		59	DO	00089	6\$:	MOVL	START_CNT, I
				1D	11	0008C		BRB	10\$
50		58		56	C3	0008E	7\$:	SUBL3	I, LOC_FRAME_WIDTH, R0
		50		6A	D1	00092		CMPL	(R10), R0
				05	19	00095		BLSS	8\$
		6E		50	DO	00097		MOVL	R0, NUM_CHARS
				03	11	0009A		BRB	9\$
04	BE	6E		6A	DO	0009C	8\$:	MOVL	(R10), NUM_CHARS
		6B		6E	28	0009F	9\$:	MOVC3	NUM_CHARS, (TEMP_PTR), APTR
		04	AE	6E	CO	000A4		ADDL2	NUM_CHARS, PTR
		56		6A	CO	000A8		ADDL2	(R10), I
		58		56	D1	000AB	10\$:	CMPL	I, LOC_FRAME_WIDTH
				DE	15	000AE		BLEQ	7\$
9A		57	0C	AE	F2	000B0	11\$:	A0BLSS	LOC_FRAME_LENGTH, L, 5\$
				04	000B5			RET	

; Routine Size: 182 bytes, Routine Base: CODE + 2CF7

```
: 5219      6123 1 %sbttl 'MOVE_FRAME - Move Information into this Frame of the Page'
: 5220      6124 1 ++
: 5221      6125 1 Functional Description:
: 5222      6126 1 This procedure inserts a string(frame) into an array(frame).
: 5223      6127 1 Insertion continues until either no more string or no more room.
: 5224      6128 1
: 5225      6129 1 Formal Parameters:
: 5226      6130 1 SCB          - Address of the SCB
: 5227      6131 1 CHAR_STRING - Descriptor of String to Insert
: 5228      6132 1 FRAME_PTR   - Address of first byte of Frame
: 5229      6133 1 FRAME_LENGTH - Length of Frame
: 5230      6134 1 FRAME_WIDTH  - Width of Frame
: 5231      6135 1
: 5232      6136 1 Implicit Inputs:
: 5233      6137 1             none
: 5234      6138 1
: 5235      6139 1 Implicit Outputs:
: 5236      6140 1             none
: 5237      6141 1
: 5238      6142 1 Returned Value:
: 5239      6143 1             none
: 5240      6144 1
: 5241      6145 1 Side Effects:
: 5242      6146 1             Truncation is possible.
: 5243      6147 1
: 5244      6148 1 --
: 5245      6149 1 ROUTINE MOVE_FRAME (
: 5246      6150 1     SCB          : REF $BBLOCK
: 5247      6151 1     CHAR_STRING : REF VECTOR[2],
: 5248      6152 1     FRAME_PTR   : REF PAGE_ARRAY
: 5249      6153 1     FRAME_WIDTH  :
: 5250      6154 1     FRAME_LENGTH :               ! Number of Columns
: 5251      6155 1     ) : NOVALUE =           ! Number of Rows
: 5252      6156 2 BEGIN
: 5253      6157 2
: 5254      6158 2 LOCAL  PTR          : REF PAGE_ARRAY,
: 5255      6159 2             LOC_FRAME_LENGTH,
: 5256      6160 2             LOC_FRAME_WIDTH ,
: 5257      6161 2             STR_PTR
: 5258      6162 2             CURR_SIZE
: 5259      6163 2             NUM_CHARS;
: 5260      6164 2
: 5261      6165 2 ! Check for dumb calls
: 5262      6166 2
: 5263      6167 3 IF (.FRAME_LENGTH LEQ 0) OR (.FRAME_WIDTH LEQ 0)
: 5264      6168 2 THEN
: 5265      6169 2     RETURN;
: 5266      6170 2 IF (.FRAME_LENGTH GTR .SCB[PSMSL_PAGE_LENGTH]) OR
: 5267      6171 3     (.FRAME_WIDTH GTR .SCB[PSMSL_PAGE_WIDTH])
: 5268      6172 2 THEN
: 5269      6173 2     RETURN;
: 5270      6174 2
: 5271      6175 2
: 5272      6176 2 ! Check page boundary conditions
: 5273      6177 2
: 5274      6178 2 LOC_FRAME_LENGTH = .FRAME_LENGTH;
: 5275      6179 2 IF .LOC_FRAME_LENGTH GTR .SCB[PSMSL_PAGE_LENGTH]
```

```

: 5276
: 5277
: 5278
: 5279
: 5280
: 5281
: 5282
: 5283
: 5284
: 5285
: 5286
: 5287
: 5288
: 5289
: 5290
: 5291
: 5292
: 5293
: 5294
: 5295
: 5296
: 5297
: 5298
: 5299
: 5300
: 5301
: 5302
: 5303
: 5304
: 5305
: 5306
: 5307
: 5308
: 5309
: 5310
: 5311
: 5312
: 5313
: 5314

      6180 2 THEN
      6181    LOC_FRAME_LENGTH = .SCB[PSMSL_PAGE_LENGTH];           ! stay in page bounds
      6182
      6183    LOC_FRAME_WIDTH = .FRAME_WIDTH;
      6184    IF .LOC_FRAME_WIDTH GTR .SCB[PSMSL_PAGE_WIDTH]
      6185    THEN
      6186        LOC_FRAME_WIDTH = .SCB[PSMSL_PAGE_WIDTH];           ! stay in page bounds
      6187
      6188    ! Get string info
      6189    CURR_SIZE          = .CHAR_STRING[SIZE];
      6190    PTR                 = FRAME_PTR[0,0,.SCB[PSMSL_PAGE_WIDTH]];
      6191    STR_PTR             = .CHAR_STRING[ADDR];
      6192
      6193    ! Do a quick fill of the frame
      6194    FILL_FRAME (.SCB,
      6195        XCHAR(32),           ! fill with blanks
      6196        FRAME_PTR[0,0,.SCB[PSMSL_PAGE_WIDTH]],
      6197        .LOC_FRAME_WIDTH,
      6198        .LOC_FRAME_LENGTH);
      6199
      6200    DECRU L FROM .LOC_FRAME_LENGTH TO 1 DO
      6201        BEGIN
      6202            IF .CURR_SIZE GEQ .LOC_FRAME_WIDTH THEN
      6203                NUM_CHARS = .LOC_FRAME_WIDTH
      6204            ELSE
      6205                NUM_CHARS = .CURR_SIZE;
      6206
      6207                CH$MOVE(.NUM_CHARS, .STR_PTR, .PTR);
      6208
      6209                PTR = .PTR + .SCB[PSMSL_PAGE_WIDTH];           ! Address calc. is based
      6210
      6211                STR_PTR = .STR_PTR + .LOC_FRAME_WIDTH;
      6212
      6213                IF .LOC_FRAME_WIDTH GTRU .CURR_SIZE THEN
      6214                    EXITLOOP;
      6215
      6216                CURR_SIZE = .CURR_SIZE - .NUM_CHARS;           ! Decrease string size
      6217
      6218    END;
      6219 1 END;

```

OFFC 00000 MOVE_FRAME:						
5E	14	04 C2 00002	.WORD	Save R2,R3,R4,R5,R6,R7,R8,R9,R10,R11		6149
51		AC D0 00005	SUBL2	#4, SP		6167
	10	7C 15 00009	MOVL	FRAME_LENGTH, R1		
			BLEQ	7\$		
57	04	AC D5 0000B	TSTL	FRAME_WIDTH		
50	01F8	77 15 0000E	BLEQ	7\$		
50		AC D0 00010	MOVL	SCB, R7		6170
		C7 D0 00014	MOVL	504(R7), R0		
		51 D1 00019	CMPL	R1, R0		
0200	C7	69 14 0001C	BGTR	7\$		
		AC D1 0001E	CMPL	FRAME_WIDTH, 512(R7)		6171
		61 14 00024	BGTR	7\$		

SEPARATE
V04-001

Print Symbiont -- separation routines
MOVE_FRAME - Move Information into th

L 10

16-Sep-1984 02:23:03
14-Sep-1984 22:32:26

VAX-11 Bliss-32 V4.0-742
[PRTSMB.SRC]SEPARATE.B32:2

Page 176
(44)

SE
VO

; Routine Size: 136 bytes, Routine Base: CODE + 2DAD

6219 1 %sbttl 'INSERT_FRAME - Insert Information into this Frame of the Page'
6220 1 ++
6221 1 Functional Description:
6222 1 This procedure inserts a string into an array(frame).
6223 1 Insertion continues until either no more string or no more room.
6224 1 Delimiting characters are used to correctly parse the string
6225 1 prior to insertion.
6226 1
6227 1 Formal Parameters:
6228 1 SCB - Address of the SCB
6229 1 CHAR_STRING - Descriptor of String to Insert
6230 1 FRAME_PTR - Address of first byte of Frame
6231 1 FRAME_LENGTH - Length of Frame
6232 1 FRAME_WIDTH - Width of Frame
6233 1
6234 1 Implicit Inputs:
6235 1 none
6236 1
6237 1 Implicit Outputs:
6238 1 none
6239 1
6240 1 Returned Value:
6241 1 none
6242 1
6243 1 Side Effects:
6244 1 Truncation is possible.
6245 1
6246 1 !--
6247 1 ROUTINE INSERT_FRAME {
6248 1 SCB : REF \$BBLOCK,
6249 1 CHAR_STRING : REF VECTOR[2],
6250 1 FRAME_PTR : REF PAGE_ARRAY,
6251 1 FRAME_WIDTH . ! Number of Columns
6252 1 FRAME_LENGTH . ! Number of Rows
6253 1) : NOVALUE =
6254 2 BEGIN
6255 2 BUILTIN AP; ! just in case truncation occurred ... don't delimit
6256 2
6257 2 LITERAL
6258 2 LEADING = 0;
6259 2 TRAILING = 1;
6260 2
6261 2 LOCAL PTR : REF PAGE_ARRAY,
6262 2 LOC_FRAME_LENGTH,
6263 2 LOC_FRAME_WIDTH ,
6264 2 STR_PTR .
6265 2 CURR_SIZE .
6266 2 DUM_EN .
6267 2 NUM_CHARS;
6268 2
6269 2 ! Check for dumb calls
6270 2
6271 3 IF (.FRAME_LENGTH LEQ 0) OR (.FRAME_WIDTH LEQ 0)
6272 2 THEN
6273 2 RETURN;
6274 2 IF (.FRAME_LENGTH GTR .SCB[PSMSL_PAGE_LENGTH]) OR
6275 3 (.FRAME_WIDTH GTR .SCB[PSMSL_PAGE_WIDTH])

N 10
16-Sep-1984 02:23:03
14-Sep-1984 22:32:26
VAX-11 Bliss-32 v4.0-742
[PRTSMB.SRC]SEPARATE.B32;2

```
Print Symbiont -- separation routines
INSERT_FRAME - Insert Information into this Fra

6276 2 THEN
6277 2 RETURN;
6278 2 ! Check page boundary conditions
6279 2 LOC_FRAME_LENGTH = .FRAME_LENGTH;
6280 2 IF .LOC_FRAME_LENGTH GTR .SCB[PSMSL_PAGE_LENGTH]
6281 2 THEN
6282 2 LOC_FRAME_LENGTH = .SCB[PSMSL_PAGE_LENGTH];           ! stay in page bounds
6283 2 LOC_FRAME_WIDTH = .FRAME_WIDTH;
6284 2 IF .LOC_FRAME_WIDTH GTR .SCB[PSMSL_PAGE_WIDTH]
6285 2 THEN
6286 2 LOC_FRAME_WIDTH = .SCB[PSMSL_PAGE_WIDTH];           ! stay in page bounds
6287 2 ! Get string info
6288 2 CURR_SIZE      = .CHAR_STRING[SIZE];
6289 2 PTR            = FRAME_PTR[0,0,.SCB[PSMSL_PAGE_WIDTH]];
6290 2 STR_PTR        = .CHAR_STRING[ADDR];
6291 2 ! Do a quick fill of the frame
6292 2 FILL_FRAME (.SCB,
6293 2                 %CHAR(32),          ! fill with blanks
6294 2                 FRAME_PTR[0,0,.SCB[PSMSL_PAGE_WIDTH]],
6295 2                 .LOC_FRAME_WIDTH,
6296 2                 .LOC_FRAME_LENGTH);
6297 2 DECR L FROM .LOC_FRAME_LENGTH TO 1 DO
6298 2 BEGIN
6299 2     IF .CURR_SIZE GEQ .LOC_FRAME_WIDTH THEN
6300 2         NUM_CHARS = DELIMIT_STRING(.STR_PTR,%CHAR(32),.LOC_FRAME_WIDTH)
6301 2     ELSE
6302 2         NUM_CHARS = .CURR_SIZE;
6303 2     DISCARD (LEADING, %C' ', .STR_PTR, .NUM_CHARS, .NUM_CHARS, .STR_PTR);
6304 2     CH$MOVE (.NUM_CHARS, .STR_PTR, .PTR);
6305 2     PTR = .PTR + .SCB[PSMSL_PAGE_WIDTH];                  ! Address calc. is based
6306 2     STR_PTR = .STR_PTR + .NUM_CHARS;
6307 2     IF .LOC_FRAME_WIDTH GEQ .CURR_SIZE THEN
6308 2         EXITLOOP;
6309 2     CURR_SIZE = .CURR_SIZE - .NUM_CHARS;                  ! Decrease string size
6310 2     END;
6311 2     IF .STR_PTR LSS (.CHAR_STRING[ADDR] + .CHAR_STRING[SIZE])
6312 2     THEN
6313 2         ! truncation occurred
6314 2         CALLG (.AP, MOVE_FRAME);                         ! dont delimit..just move str.
6315 2     END;
```

				WORD	Save R2,R3,R4,R5,R6,R7,R8,R9,R10,R11	: 6247
	5E	14	08 C2 00002	SUBL2	#8, SP	: 6271
			AC D0 00005	MOVL	FRAME_LENGTH, R1	
		10	03 15 00009	BLEQ	1\$	
			AC D5 00008	TSTL	FRAME_WIDTH	
			01 14 0000E	BGTR	2\$	
			04 00010	RET		
	58	04	AC D0 00011	MOVL	SCB, R8	: 6274
	50	01F8	C8 D0 00015	MOVL	504(R8), R0	
	50		51 D1 0001A	CMPL	R1, R0	
0200	C8	10	06 14 0001D	BGTR	3\$	
			AC D1 0001F	CMPL	FRAME_WIDTH, 512(R8)	: 6275
			01 15 00025	BLEQ	4\$	
			04 00027	RET		
	52		51 D0 00028	MOVL	R1, LOC FRAME LENGTH	: 6281
	50		52 D1 0002B	CMPL	LOC_FRAME_LENGTH, R0	: 6282
			03 15 0002E	BLEQ	5\$	
	52		50 D0 00030	MOVL	R0, LOC FRAME LENGTH	: 6284
0200	C8	10	59 D0 00033	MOVL	FRAME_WIDTH, [OC_FRAME WIDTH	: 6286
			59 D1 00037	CMPL	LOC_FRAME_WIDTH, 512(R8)	: 6287
			05 15 0003C	BLEQ	6\$	
	59	0200	C8 D0 0003E	MOVL	512(R8), LOC_FRAME_WIDTH	: 6289
	57	08	AC D0 00043	MOVL	CHAR_STRING, R7	: 6292
	5A		67 D0 00047	MOVL	(R7)- CURR_SIZE	
	5B	0C	AC D0 0004A	MOVL	FRAME_PTR, PTR	: 6293
	6E	04	A7 D0 0004E	MOVL	4(R7), STR PTR	: 6294
			52 DD 00052	PUSHL	LOC_FRAME_LENGTH	: 6301
			59 DD 00054	PUSHL	LOC_FRAME_WIDTH	: 6300
		0C	AC DD 00056	PUSHL	FRAME_PTR	: 6299
			20 DD 00059	PUSHL	#32	
			58 DD 0005B	PUSHL	R8	
FEOB	CF	01	05 FB 0005D	CALLS	#5, FILL_FRAME	
	56		A2 9E 00062	MOVAB	1(R2), L	: 6303
			47 11 00066	BRB	10\$	
	59		5A D1 00068	CMPL	CURR_SIZE, LOC_FRAME_WIDTH	: 6305
			12 19 0006B	BLSS	8\$	
			59 DD 0006D	PUSHL	LOC_FRAME_WIDTH	: 6306
			20 DD 0006F	PUSHL	#32	
		08	AE DD 00071	PUSHL	STR_PTR	
0000V	CF		03 FB 00074	CALLS	#3,-DELIMIT STRING	
04	AE		50 D0 00079	MOVL	R0, NUM_CHARS	
			04 11 0007D	BRB	9\$	
04	AE		5A D0 0007F	MOVL	CURR_SIZE, NUM_CHARS	: 6308
			5E DD 00083	PUSHL	SP	: 6310
		08	AE 9F 00085	PUSHAB	NUM_CHARS	
		0C	AE DD 00088	PUSHL	NUM_CHARS	
		0C	AE DD 0008B	PUSHL	STR_PTR	
			20 DD 0008E	PUSHL	#32	
			7E D4 00090	CLRL	-(SP)	
6B	0000V	CF	06 FB 00092	CALLS	#6, DISCARD	
	50	04	AE 28 00097	MOVC3	NUM_CHARS, ASTR_PTR, (PTR)	: 6311
	5B	0200	C8 C0 0009D	ADDL2	512(R8), PTR	: 6313
	6E	04	AE C0 000A2	ADDL2	NUM_CHARS, STR_PTR	: 6315
	5A		59 D1 000A6	CMPL	LOC_FRAME_WIDTH, CURR_SIZE	: 6317
			07 18 000A9	BGEQ	11\$	
	5A	04	AE C2 000AB	SUBL2	NUM_CHARS, CURR_SIZE	: 6320
			B6 F5 000AF	S0BGTR	L, 7\$: 6303

SEPARATE
V04-001

Print Symbiont -- separation routines
INSERT_FRAME - Insert Information into this Fra

{ 11

16-Sep-1984 02:23:03

14-Sep-1984 22:32:26

VAX-11 Bliss-32 V4.0-742
[PRTSMB.SRC]SEPARATE.B32;2

Page 180
(45)

SE
VO

57	04	A7	67	C1	000B2	11\$:	ADDL3	(R7), 4(R7), R7
		57	6E	D1	000B7		CMPL	STR_PTR, R7
			05	18	000BA		BGEQ	12\$
FEB7	CF		6C	FA	000BC		CALLG	(AP), MOVE_FRAME
			04	000C1	12\$:		RET	

: 6323
: 6325
: 6326

; Routine Size: 194 bytes, Routine Base: CODE + 2E35

: 5425 6327 1 %sbttl 'CENTER_FRAME - Insert String Information into the Center of this Frame'
: 5426 6328 1 ++
: 5427 6329 1 Functional Description:
: 5428 6330 1 This procedure inserts a string into the center of an array(frame).
: 5429 6331 1 Insertion continues until either no more string or no more room.
: 5430 6332 1 The idea is to center the string within the frame.
: 5431 6333 1 1) If the string is shorter than the entire frame then
: 5432 6334 1 center the string in the frame.
: 5433 6335 1 2) If the string is longer than the entire frame then
: 5434 6336 1 insert as much of the string as possible.
: 5435 6337 1 3) Use an assumed pad of blanks for beginning and end of string.
: 5436 6338 1
: 5437 6339 1
: 5438 6340 1 Formal Parameters:
: 5439 6341 1 SCB - Address of the SCB
: 5440 6342 1 CHAR_STRING - Descriptor of String to Insert
: 5441 6343 1 FRAME_PTR - Address of first byte of Frame
: 5442 6344 1 FRAME_LENGTH - Length of Frame
: 5443 6345 1 FRAME_WIDTH - Width of Frame
: 5444 6346 1
: 5445 6347 1 Implicit Inputs:
: 5446 6348 1 none
: 5447 6349 1
: 5448 6350 1 Implicit Outputs:
: 5449 6351 1 none
: 5450 6352 1
: 5451 6353 1 Returned Value:
: 5452 6354 1 none
: 5453 6355 1
: 5454 6356 1 Side Effects:
: 5455 6357 1 Truncation is possible.
: 5456 6358 1
: 5457 6359 1 --
: 5458 6360 1 ROUTINE CENTER_FRAME {
: 5459 6361 1 SCB : REF \$BBLOCK,
: 5460 6362 1 CHAR_STRING : REF VECTOR[2],
: 5461 6363 1 FRAME_PTR : REF PAGE_ARRAY,
: 5462 6364 1 FRAME_WIDTH : Number of Columns
: 5463 6365 1 FRAME_LENGTH : Number of Rows
: 5464 6366 1) : NOVALUE =
: 5465 6367 2 BEGIN
: 5466 6368 2 LITERAL K_MAX_BUFFER_SIZE = 512. ! maximum possible buffer size
: 5467 6369 2 K_PAD_LEN = 2; ! length to pad the string
: 5468 6370 2
: 5469 6371 2 LOCAL PTR : REF PAGE_ARRAY, ! points to col 0, variable row
: 5470 6372 2 LOC_FRAME_LENGTH, ! local count of frame_length
: 5471 6373 2 LOC_FRAME_WIDTH, ! local count of frame_width
: 5472 6374 2 PAD_CHAR : ! pad character is assumed a space
: 5473 6375 2 PTR_OFFSET : SIGNED, ! pos/neg offset to origin
: 5474 6376 2 STR_PTR : ! points to str position in frame
: 5475 6377 2 STR_SIZE : ! num char left in string
: 5476 6378 2 BUFFER : VECTOR [512,byte], ! variable buff for pad and string
: 5477 6379 2 STR_DESC : VECTOR [2]; ! desc of string
: 5478 6380 2
: 5479 6381 2 ! Check for dumb calls. Frame dimensions must be writable
: 5480 6382 2
: 5481 6383 3 IF (.FRAME_LENGTH LEQ 0) OR (.FRAME_WIDTH LEQ 0)

```
; 5482    6384 2 THEN
; 5483    6385 2 RETURN:
; 5484    6386 2 IF (.FRAME_LENGTH GTR .SCB[PSMSL_PAGE_LENGTH]) OR
; 5485    6387 2 (.FRAME_WIDTH GTR .SCB[PSMSL_PAGE_WIDTH])
; 5486    6388 2 THEN
; 5487    6389 2 RETURN:
; 5488    6390 2 ! String must not be zero !!
; 5489    6391 2
; 5490    6392 2 IF .CHAR_STRING[SIZE] EQL 0 THEN RETURN SSS_NORMAL;
; 5491    6393 2
; 5492    6394 2 ! Check page boundary conditions
; 5493    6395 2
; 5494    6396 2 LOC_FRAME_LENGTH = .FRAME_LENGTH;
; 5495    6397 2 IF .LOC_FRAME_LENGTH GTR .SCB[PSMSL_PAGE_LENGTH]
; 5496    6398 2 THEN
; 5497    6399 2     LOC_FRAME_LENGTH = .SCB[PSMSL_PAGE_LENGTH];      ! stay in page bounds
; 5498    6400 2
; 5499    6401 2 LOC_FRAME_WIDTH = .FRAME_WIDTH;
; 5500    6402 2 IF .LOC_FRAME_WIDTH GTR .SCB[PSMSL_PAGE_WIDTH]
; 5501    6403 2 THEN
; 5502    6404 2     LOC_FRAME_WIDTH = .SCB[PSMSL_PAGE_WIDTH];      ! stay in page bounds
; 5503    6405 2
; 5504    6406 2 ! Get string into padding buffer if enough room
; 5505    6407 2
; 5506    6408 2 STR_DESC[SIZE] = %ALLOCATION(BUFFER);   ! length of string and pad chars
; 5507    6409 2 STR_DESC[ADDR] = BUFFER;                  ! ptr into pad&string buffer
; 5508    6410 2 PAD_CHAR = %CHAR(32,32);
; 5509    6411 2 PTR = FRAME_PTR[0,0,.SCB[PSMSL_PAGE_WIDTH]]; ! init
; 5510    6412 2
; 5511    6413 2 ! Pad the string if there is enough room
; 5512    6414 3 IF (.CHAR_STRING[SIZE] + (2 * K_PAD_LEN))
; 5513    6415 3     LEO
; 5514    6416 3     (.LOC_FRAME_WIDTH * .LOC_FRAME_LENGTH)
; 5515    6417 2 THEN
; 5516    6418 3 BEGIN
; 5517    6419 3     ! Set the size to correct value
; 5518    6420 3
; 5519    6421 3     IF .CHAR_STRING[SIZE] LEQ (K_MAX_BUFFER_SIZE-2*(K_PAD_LEN)) THEN
; 5520    6422 4     STR_DESC[SIZE] = .CHAR_STRING[SIZE] + (2 * (R_PAD_LEN))
; 5521    6423 3     ELSE
; 5522    6424 3     STR_DESC[SIZE] = K_MAX_BUFFER_SIZE;
; 5523    6425 3
; 5524    6426 3     ! Pad the string using the local buffer
; 5525    6427 3     CH$COPY(K_PAD_LEN, PAD_CHAR, .CHAR_STRING[SIZE], .CHAR_STRING[ADDR],
; 5526    6428 3             K_PAD_LEN, PAD_CHAR, .PAD_CHAR, .STR_DESC[SIZE],
; 5527    6429 3             .STR_DESC[ADDR]);
; 5528    6430 3 END
; 5529    6431 2 ELSE
; 5530    6432 3     ! copy into local buffer
; 5531    6433 3     BEGIN
; 5532    6434 3     CH$MOVE(.CHAR_STRING[SIZE], .CHAR_STRING[ADDR], .STR_DESC[ADDR]);
; 5533    6435 3     STR_DESC[SIZE] = .CHAR_STRING[SIZE];
; 5534    6436 2 END;
; 5535    6437 2 ! Calc offset to pointer using new padded length
; 5536    6438 3 IF (.LOC_FRAME_WIDTH * .LOC_FRAME_LENGTH) GTR (.STR_DESC[SIZE])
; 5537    6439 2 THEN
; 5538    6440 4     PTR_OFFSET = ((.LOC_FRAME_WIDTH * .LOC_FRAME_LENGTH)
```

```

: 5539      6441 2           = (.STR_DESC[SIZE])/2
: 5540      6442 2 ELSE
: 5541      6443 2   PTR_OFFSET = 0;
: 5542      6444
: 5543      6445 2 ! Check for negative offset
: 5544      6446 2 IF .PTR_OFFSET LSS 0
: 5545      6447 2 THEN
: 5546      6448 2   PTR_OFFSET = 0;
: 5547      6449 2
: 5548      6450 2 ! Set pointer to buffer
: 5549      6451 2 STR_PTR = .PTR + .PTR_OFFSET;
: 5550      6452 2 STR_SIZE = .STR_DESC[SIZE];
: 5551      6453 2
: 5552      6454 2 DECRU L FROM .LOC_FRAME_LENGTH TO 1 DO
: 5553      6455 3 BEGIN
: 5554      6456 3
: 5555      6457 3 LOCAL NUM_CHARS; ! number of chars to move to the frame
: 5556      6458 3
: 5557      6459 3 IF .PTR_OFFSET GEQ .LOC_FRAME_WIDTH THEN
: 5558      6460 4 BEGIN
: 5559      6461 4   PTR = .PTR + .SCB[PSMSL_PAGE_WIDTH]; ! go to next row of frame
: 5560      6462 4   PTR_OFFSET = .PTR_OFFSET - .LOC_FRAME_WIDTH; ! adjust offset to column
: 5561      6463 4 END
: 5562      6464 4 ELSE BEGIN
: 5563      6465 4
: 5564      6466 4   IF .STR_SIZE GEQ (.LOC_FRAME_WIDTH - .PTR_OFFSET) THEN
: 5565      6467 4     NUM_CHARS = .LOC_FRAME_WIDTH - .PTR_OFFSET
: 5566      6468 4   ELSE ! check for overflow of frame width
: 5567      6469 4     NUM_CHARS = .STR_SIZE; ! - insert which ever is less
: 5568      6470 4
: 5569      6471 4 CHSMOVE(.NUM_CHARS, .STR_DESC[ADDR], .STR_PTR);
: 5570      6472 4 STR_PTR = .PTR + .SCB[PSMSL_PAGE_WIDTH]; ! Address calc. is base on
: 5571      6473 4 PTR = .STR_PTR; ! Frame ptr[0,0] & Form Width
: 5572      6474 4
: 5573      6475 4 STR_DESC[ADDR] = .STR_DESC[ADDR] + .NUM_CHARS;
: 5574      6476 4
: 5575      6477 4 IF .LOC_FRAME_WIDTH GTRU .STR_SIZE ! Already inserted it all
: 5576      6478 4 THEN EXITLOOP;
: 5577      6479 4
: 5578      6480 4
: 5579      6481 4 STR_SIZE = .STR_SIZE - .NUM_CHARS; ! Decrease string size
: 5580      6482 4
: 5581      6483 4 ! reset the offset to start column one
: 5582      6484 4 PTR_OFFSET = 0;
: 5583      6485 4
: 5584      6486 3 END;
: 5585      6487 2 END;
: 5586      6488 1 END;

```

OFFC 00000 CENTER_FRAME:

5E	FDE8	CE	9E	00002	.WORD	Save R2,R3,R4,R5,R6,R7,R8,R9,R10,R11
52	14	AC	D0	00007	MOVAB	-536(SP), SP
					MOVL	FRAME_LENGTH, R2

: 6360

: 6383

SEPARATE
V04-001

Print_Symbiont -- separation routines
CENTER_FRAME - Insert String Information

6 11

16-Sep-1984 02:23:03
14-Sep-1984 22:32:26

VAX-11 Bliss-32 V4.0-742
[PRTSMB.SRC]SEPARATE.B32;2

Page 184
(46)

SEPARATE
V04-001Print Symbiont -- separation routines
CENTER_FRAME - Insert String Information into t

H 11

16-Sep-1984 02:23:03
14-Sep-1984 22:32:26VAX-11 Bliss-32 v4.0-742
[PRTSMB.SRC]SEPARATE.B32;2Page 185
(46)SE
VO

5A	50		02 C7 000D3	DIVL3	#2 R0, PTR_OFFSET	
			02 11 000D7	BRB	13\$ PTR_OFFSET	6440
			5A D4 000D9	12\$:	CLRL PTR_OFFSET	6443
			02 18 000DB	13\$:	BGEQ 14\$ PTR_OFFSET	6446
			5A D4 000DD	14\$:	CLRL PTR_OFFSET	6448
5B	5A	08	AE C1 000DF	ADDL3	PTR, PTR_OFFSET, STR_PTR	6451
	59	10	AE D0 000E4	MOVL	STR_DESC, STR_SIZE	6452
			3F 11 000E8	BRB	20\$ PTR_OFFSET	6454
			5A D1 000EA	15\$:	CMPL PTR_OFFSET, LOC_FRAME_WIDTH	6459
			0A 19 000ED	BLSS	16\$	
	08	AE	BE C0 000EF	ADDL2	#4(SP), PTR	6461
			57 C2 000F4	SUBL2	LOC_FRAME_WIDTH, PTR_OFFSET	6462
			2E 11 000F7	BRB	19\$	6459
50	57		5A C3 000F9	16\$:	PTR_OFFSET, LOC_FRAME_WIDTH, R0	6466
	50		59 D1 000FD	CMPL	STR_SIZE, R0	
			05 19 00100	BLSS	17\$	
			50 D0 00102	MOVL	R0, NUM_CHARS	6467
			03 11 00105	BRB	18\$	
	58		59 D0 00107	17\$:	MOVL STR_SIZE, NUM_CHARS	6469
6B	14	BE	58 28 0010A	MOV C3	NUM_CHARS, ASTR DESC+4, (STR_PTR)	6471
5B	08	AE	18\$:	ADDL3	#4(SP), PTR, STR_PTR	6472
	08	AE	BE C1 0010F	MOVL	STR_PTR, PTR	6473
	14	AE	5B D0 00115	ADDL2	NUM_CHARS, STR_DESC+4	6475
			58 C0 00119	CMPL	LOC_FRAME_WIDTH, STR_SIZE	6477
			57 D1 0011D	BGTRU	21\$	
			0B 1A 00120	SUBL2	NUM_CHARS, STR_SIZE	6481
			58 C2 00122	CLRL	PTR_OFFSET	6484
	59		5A D4 00125	DECL	L	6454
			56 D7 00127	19\$:	TSTL	
			56 D5 00129	20\$:	L	
			BD 12 0012B	BNEQ	15\$	
			04 0012D	21\$:	RET	6488

; Routine Size: 302 bytes. Routine Base: CODE + 2EF7

: 5588 6489 1 %sbttl 'MERGE_FRAME - Merge Information in this Frame of the Page'
: 5589 6490 1 ++
: 5590 6491 1 Functional Description:
: 5591 6492 1 This procedure merges rows of non-blank strings into an array(frame).
: 5592 6493 1 Merging continues until either no more strings or no more frame. The
: 5593 6494 1 contents of the frame are merged to the base of the frame. Any row
: 5594 6495 1 with data present is considered impure and is merged.
: 5595 6496 1
: 5596 6497 1 Formal Parameters:
: 5597 6498 1 SCB - Address of the SCB
: 5598 6499 1 CHAR_STRING - Descriptor of String to Insert
: 5599 6500 1 FRAME_PTR - Address of first byte of Frame
: 5600 6501 1 FRAME_LENGTH - Length of Frame
: 5601 6502 1 FRAME_WIDTH - Width of Frame
: 5602 6503 1
: 5603 6504 1 Implicit Inputs:
: 5604 6505 1 none
: 5605 6506 1
: 5606 6507 1 Implicit Outputs:
: 5607 6508 1 none
: 5608 6509 1
: 5609 6510 1 Returned Value:
: 5610 6511 1 none
: 5611 6512 1
: 5612 6513 1 Side Effects:
: 5613 6514 1 Truncation is possible.
: 5614 6515 1
: 5615 6516 1 --
: 5616 6517 1 ROUTINE MERGE_FRAME (
: 5617 6518 1 SCB : REF \$BBLOCK,
: 5618 6519 1 CHAR_STRING : REF VECTOR[2],
: 5619 6520 1 FRAME_PTR : REF PAGE_ARRAY,
: 5620 6521 1 FRAME_WIDTH : Number of Columns
: 5621 6522 1 FRAME_LENGTH : Number of Rows
: 5622 6523 1 RET_LEN : REF VECTOR[,word]
: 5623 6524 1) : NOVALUE =
: 5624 6525 2 BEGIN
: 5625 6526 2 LITERAL K_MAX_SIZE = 256;
: 5626 6527 2 LOCAL
: 5627 6528 2 CLR_STR : VECTOR[2],
: 5628 6529 2 SRCE_STR : VECTOR[2],
: 5629 6530 2 BUFFER : VECTOR[256,byte],
: 5630 6531 2 CURRENT_PTR : REF PAGE_ARRAY,
: 5631 6532 2 CURRENT_LEN :
: 5632 6533 2 DEST_OFFSET :
: 5633 6534 2 SOURCE_OFFSET :
: 5634 6535 2
: 5635 6536 2
: 5636 6537 2
: 5637 6538 2 : setup clr_str
: 5638 6539 2 :
: 5639 6540 2 CLR_STR[ADDR] = .CHAR_STRING[ADDR];
: 5640 6541 2 CLR_STR[SIZE] = .FRAME_WIDTH;
: 5641 6542 2 CHSFILL ('C', .FRAME_WIDTH, .CLR_STR[ADDR]);
: 5642 6543 2 : setup source_str
: 5643 6544 2 :
: 5644 6545 2 SRCE_STR[SIZE] = XALLOCATION(BUFFER);

```
: 5645      6546 2 SRCE_STR[ADDR] = BUFFER;
: 5646      6547 2
: 5647      6548 2 | setup current point and offsets into frame
: 5648      6549 2
: 5649      6550 2 CURRENT_PTR = FRAME_PTR[0,0, .SCB[PSMSL_PAGE_WIDTH]];
: 5650      6551 2
: 5651      6552 2 SOURCE_OFFSET = .FRAME_LENGTH-1;           ! start at frame boundaries
: 5652      6553 2 DEST_OFFSET = .FRAME_LENGTH-1;
: 5653      6554 2
: 5654      6555 2 DECRU I FROM (.FRAME_LENGTH-1) TO 0
: 5655      6556 2 DO
: 5656      6557 2     BEGIN
: 5657      6558 2     FIND_DEST_PTR (
: 5658      6559 2         .SCB,
: 5659      6560 2         CLR_STR[0],
: 5660      6561 2         CURRENT_PTR[0,0,.SCB[PSMSL_PAGE_WIDTH]],
: 5661      6562 2         .FRAME_WIDTH,
: 5662      6563 2         .FRAME_LENGTH,
: 5663      6564 2         DEST_OFFSET);
: 5664      6565 2     FIND_SOURCE_PTR (
: 5665      6566 2         .SCB,
: 5666      6567 2         SRCE_STR[0],
: 5667      6568 2         CURRENT_PTR[0,0,.SCB[PSMSL_PAGE_WIDTH]],
: 5668      6569 2         .FRAME_WIDTH,
: 5669      6570 2         .DEST_OFFSET,
: 5670      6571 2         SOURCE_OFFSET);
: 5671      6572 2     ! Exit loop when no source string
: 5672      6573 2     ! IF .SRCE_STR[SIZE] EQ 0
: 5673      6574 2     THEN
: 5674      6575 2     BEGIN
: 5675      6576 2     RET_LEN[0] = .FRAME_LENGTH - .I;
: 5676      6577 2     RETURN;
: 5677      6578 2     END;
: 5678      6579 2
: 5679      6580 2     ! Move the source to the destination
: 5680      6581 2
: 5681      6582 2     MOVE_FRAME (
: 5682      6583 2         .SCB,
: 5683      6584 2         SRCE_STR[0],
: 5684      6585 2         CURRENT_PTR[0,.DEST_OFFSET,.SCB[PSMSL_PAGE_WIDTH]],          ! string frame reference
: 5685      6586 2
: 5686      6587 2         .FRAME_WIDTH,                                     ! ref-to-frame
: 5687      6588 2
: 5688      6589 2         1);                                         ! cols to fill
: 5689      6590 2
: 5690      6591 2
: 5691      6592 2     ! Clear the source position
: 5692      6593 2
: 5693      6594 2     MOVE_FRAME (
: 5694      6595 2         .SCB,
: 5695      6596 2         CLR_STR[0],
: 5696      6597 2         CURRENT_PTR[0,.SOURCE_OFFSET,.SCB[PSMSL_PAGE_WIDTH]],          ! string frame reference
: 5697      6598 2
: 5698      6599 2         .FRAME_WIDTH,                                     ! ref to frame
: 5699      6600 2
: 5700      6601 2         1);                                         ! cols to fill
: 5701      6602 2
:           SRCE_STR[SIZE] = K_MAX_SIZE;
```

SEPARATE
V04-001

Print_Symbiont -- separation routines
MERGE_FRAME - Merge Information in the

K 11
16-Sep-1984 02:23:03
14-Sep-1984 22:32:26

VAX-11 Bliss-32 V4.0-742
[PRTSMB.SRC]SEPARATE.B32;2

Page 188
(47)

5702
5703

6603 2 END;
6604 1 END;

007C 00000 MERGE_FRAME:

SEPARATE
V04-001

Print Symbiont -- separation routines
MERGE_FRAME - Merge Information in this Frame o

L 11

16-Sep-1984 02:23:03
14-Sep-1984 22:32:26

VAX-11 Bliss-32 V4.0-742
[PRTSMB.SRC]SEPARATE.B32;2

Page 189
(47)

: Routine Size: 171 bytes. Routine Base: CODE + 3025

```

.: 5705    6605 1 %sbttl 'INSERT_NAME_BANNER - Insert User Name as Banner into this Frame'
.: 5706    6606 1 ++
.: 5707    6607 1 Functional Description:
.: 5708    6608 1 This procedure inserts a string into the center of an array(frame).
.: 5709    6609 1 The workhorse of this routine is the BANNER routine which creates the
.: 5710    6610 1 large letters. Insertion is attempted in the center of the frame.
.: 5711    6611 1 Otherwise, insertion starts at the left margin until no more characters
.: 5712    6612 1 will fit.
.: 5713    6613 1 NOTE: Returns the amount of space used in the frame
.: 5714    6614 1
.: 5715    6615 1 Formal Parameters:
.: 5716    6616 1 SCB           - Address of the SCB
.: 5717    6617 1 CHAR_DESC     - Desc String to Insert
.: 5718    6618 1 FRAME_PTR      - Address of first byte of Frame
.: 5719    6619 1 FRAME_LENGTH   - Length of Frame and Largest Banner
.: 5720    6620 1 FRAME_WIDTH    - Width of Frame and Height of Characters
.: 5721    6621 1 DESIRED_BAN_TYPE - Requested banner type
.: 5722    6622 1
.: 5723    6623 1 Implicit Inputs:
.: 5724    6624 1             none
.: 5725    6625 1
.: 5726    6626 1 Implicit Outputs:
.: 5727    6627 1             none
.: 5728    6628 1
.: 5729    6629 1 Returned Value:
.: 5730    6630 1             none
.: 5731    6631 1
.: 5732    6632 1 Side Effects: Truncation is possible.
.: 5733    6633 1
.: 5734    6634 1
.: 5735    6635 1 --
.: 5736    6636 1 ROUTINE INSERT_NAME_BANNER (
.: 5737    6637 1     SCB          : REF $BBLOCK
.: 5738    6638 1     CHAR_DESC    : REF VECTOR[?],
.: 5739    6639 1     FRAME_PTR    : REF PAGE_ARRAY,
.: 5740    6640 1     FRAME_WIDTH   :
.: 5741    6641 1     FRAME_LENGTH  :
.: 5742    6642 1     DESIRED_BAN_TYPE
.: 5743    6643 1             ) =
.: 5744    6644 2 BEGIN
.: 5745    6645 2 Define literals to use in "Banner" call
.: 5746    6646 2             (incl... char_repeat, line_repeat, spacing)
.: 5747    6647 2
.: 5748    6648 2 LITERAL K_LARGE LETTERS = 14,
.: 5749    6649 2 K_MAX_STRING_SIZE = 42,
.: 5750    6650 2 K_ALT_CHAR = 0,
.: 5751    6651 2 K_SPACING = 2,
.: 5752    6652 2 K_LEAD SPACES = 0,
.: 5753    6653 2 K_MAX_BUF = 512,
.: 5754    6654 2 LEAD MASK = XB'00100000',
.: 5755    6655 2 TRAILING = 1;
.: 5756    6656 2
.: 5757    6657 2 By defining local buffer and descriptor. I can call the banner routine
.: 5758    6658 2 and get the length of the string ... then use an algorithm to center the
.: 5759    6659 2 string into the frame.
.: 5760    6660 2
.: 5761    6661 2

```

! Number of Columns
! Number of Rows
! Banner size desired

! Double size chars
! max expanded chars(512 buffer)
! alternate construction char
! between character spacing
! number of leading spaces
! max for this frame buffer
! convert lower to upper case
! flag for discard
! (anything but 0 is trailing)

```
5762 6662 2 LOCAL
5763 6663 2 NUM_LINES
5764 6664 2 CHAR_REPEAT ;
5765 6665 2 LINE_REPEAT ;
5766 6666 2 BUFFER : VECTOR [512,byte];
5767 6667 2 STRING_DESC : VECTOR [2];
5768 6668 2 STR_PTR ;
5769 6669 2 STR_LEN ;
5770 6670 2 RET_LEN : VECTOR[1];
5771 6671
5772 6672 ! Dont even try if no frame
5773 6673
5774 6674 IF (.FRAME_LENGTH LSS 7) ! won't ever fit !!
5775 6675 OR (.FRAME_WIDTH LEQ 0)
5776 6676
5777 6677 THEN ! nadda...
5778 6678 RETURN 0; ! dont even try... no room...
5779 6679
5780 6680 STR_LEN = .CHAR_DESC[SIZE];
5781 6681 STR_PTR = .CHAR_DESC[ADDR];
5782 6682
5783 6683 BAS$EDIT (CHAR_DESC[0], CHAR_DESC[0], UPCASE_MASK); ! lower to upper case character
5784 6684
5785 6685 ! Insert only the string ... No trailing blanks
5786 6686
5787 6687 DISCARD (TRAILING, %C' ', .STR_PTR, .STR_LEN, ! Return length and pointer
5788 6688 STR_LEN, STR_PTR);
5789 6689
5790 6690
5791 6691 ! init the character spacing ...depends on frame_length passed in !
5792 6692 assume small chars.
5793 6693 CHAR_REPEAT = 1; ! times to repeat a char
5794 6694 LINE_REPEAT = 1; ! times to repeat a line
5795 6695 NUM_LINES = 7; ! lines equal height of banner
5796 6696
5797 6697 ! Attempt to give the caller what he wants
5798 6698 Only use large banners if they fit in the frame
5799 6699
5800 6700 IF (
5801 6701   (.DESIRED_BAN_TYPE EQL K_LARGE LETTERS) ! if he wants it
5802 6702     AND
5803 6703   ((.FRAME_LENGTH GEQ K_LARGE LETTERS) ! and...
5804 6704     AND
5805 6705   ((.FRAME_WIDTH/12) GEQ .STR_LEN)) ! if there is room !!!
5806 6706
5807 6707 THEN
5808 6708 BEGIN
5809 6709   CHAR_REPEAT = 2; ! times to repeat a char
5810 6710   LINE_REPEAT = 2; ! times to repeat a line
5811 6711   NUM_LINES = 14; ! height of banner
5812 6712 END;
5813 6713
5814 6714 ! truncate long character names to fit in buffer
5815 6715
5816 6716 IF .STR_LEN GTR K_MAX_STRING_SIZE
5817 6717 THEN .STR_LEN = K_MAX_STRING_SIZE; ! maximum 42 chars in
5818 6718
```

```

: 5819 6719 2           ! buffer of 512
: 5820 6720 2           ! get the buffer
: 5821 6721 2
: 5822 6722 2           STRING_DESC[SIZE] = %ALLOCATION(BUFFER);
: 5823 6723 2           STRING_DESC[ADDR] = BUFFER;
: 5824 6724 2
: 5825 6725 2           INCR LINE_NO FROM 0 TO (.NUM_LINES - 1)
: 5826 6726 2           DO
: 5827 6727 2           BEGIN
: 5828 6728 2           PSMSBANNER (.SCB,
: 5829 6729 2               .STR_LEN,
: 5830 6730 2               .STR_PTR,
: 5831 6731 2               K LEAD SPACES,
: 5832 6732 2               .CHAR_REPEAT,
: 5833 6733 2               .LINE_REPEAT,
: 5834 6734 2               K_SPACING,
: 5835 6735 2               K_ALT_CHAR,
: 5836 6736 2               .STRING_DESC[ADDR],
: 5837 6737 2               .LINE_NO,
: 5838 6738 2               .STRING_DESC[SIZE]);
: 5839 6739 2
: 5840 6740 2           CENTER_FRAME (.SCB,
: 5841 6741 2               STRING_DESC[0],
: 5842 6742 2               FRAME_PTR[0,.LINE_NO,.SCB[PSMBL_PAGE_WIDTH]],
: 5843 6743 2               .FRAME_WIDTH, 1);
: 5844 6744 2           ! re-init
: 5845 6745 2           STRING_DESC[SIZE] = K_MAX_BUF;          ! reset buffer size
: 5846 6746 2           END;
: 5847 6747 2
: 5848 6748 2           RETURN .NUM_LINES
: 5849 6749 1           END;

```

007C 00000 INSERT_NAME_BANNER:

				.WORD	Save R2,R3,R4,R5,R6	6636
5E	FDF0	CE 9E 00002		MOVAB	-528(SP), SP	6674
07		14 AC D1 00007		CMPL	FRAME_LENGTH, #7	
		05 19 0000B		BLSS	1\$	6676
		10 AC D5 0000D		TSTL	FRAME_WIDTH	
		03 14 00010		BGTR	2\$	
		00BD 31 00012	1\$:	BRW	7\$	
04	50	08 AC D0 00015	2\$:	MOVL	CHAR_DESC, R0	6680
	AE	60 D0 00019		MOVL	(R0), STR_LEN	
	6E	04 A0 D0 0001D		MOVL	4(R0), STR_PTR	6681
		20 DD 00021		PUSHL	#32	6683
		50 DD 00023		PUSHL	R0	
		50 DD 00025		PUSHL	R0	
		03 FB 00027		CALLS	#3, BASSEDIT	
00000000G	00	5E DD 0002E		PUSHL	SP	
		08 AE 9F 00030		PUSHAB	STR_LEN	6687
	OC	AE DD 00033		PUSHL	STR_LEN	
	OC	AE DD 00036		PUSHL	STR_PTR	
		20 DD 00039		PUSHL	#32	
		01 DD 0003B		PUSHL	#1	

SEPARATE
V04-001

Print_Symbiont -- separation routines
INSERT_NAME_BANNER - Insert User Name

C 12

16-Sep-1984 02:23:03
14-Sep-1984 22:32:26

VAX-11 Bliss-32 V4.0-742
[PRTSMB.SRC]SEPARATE.B32;2

Page 193
(48)

0000V	CF	06	FB	0003D	CALLS	#6, DISCARD	
	56	01	DO	00042	MOVL	#1, CHAR_REPEAT	6693
	55	01	DO	00045	MOVL	#1, LINE_REPEAT	6694
	54	07	DO	00048	MOVL	#7, NUM_LINES	6695
	OE	18	AC	D1 0004B	CMPL	DESIRED_BAN_TYPE, #14	6701
	OE	14	AC	D1 00051	CMPL	FRAME_LENGTH, #14	6703
50	10	AC	OC	C7 00057	DIVL3	#12, FRAME_WIDTH, R0	6705
	04	AE	50	D1 0005C	CMPL	R0, STR_LEN	
			09	19 00060	BLSS	3\$	
	56	02	DO	00062	MOVL	#2, CHAR_REPEAT	6709
	55	02	DO	00065	MOVL	#2, LINE_REPEAT	6710
	54	0E	DO	00068	MOVL	#14, NUM_LINES	6711
	2A	04	AE	D1 0006B	CMPL	STR_LEN, #42	6716
			04	15 0006F	BLEQ	4\$	
	04	AE	2A	DO 00071	MOVL	#42, STR_LEN	6718
	08	AE	0200	8F C0075	MOVZWL	#512, STRING_DESC	6722
	0C	AE	10	AE 0007B	MOVAB	BUFFR, STRING_DESC+4	6723
			04	AC 00080	MOVL	SCB, R3	6742
			01	CE 00084	MNEGL	#1, LINE_NO	
			41	11 00087	BRB	6\$	
			08	AE 9F 00089	PUSHAB	STRING_DESC	6738
			52	DD 0008C	PUSHL	LINE_NO	6737
			14	AE DD 0008E	PUSHL	STRING_DESC+4	6736
	7E		02	7D 00091	MOVO	#2, -(SP)	6728
			55	DD 00094	PUSHL	LINE_REPEAT	6733
			56	DD 00096	PUSHL	CHAR_REPEAT	6732
			7E	D4 00098	CLRL	-(SPT)	6728
			20	AE DD 0009A	PUSHL	STR_PTR	6730
			28	AE DD 0009D	PUSHL	STR_LEN	6729
			04	AC DD 000A0	PUSHL	SCB	6728
00000000G	00		08	FB 000A3	CALLS	#11, PMSBANNER	
			01	DD 000AA	PUSHL	#1	
50	52	10	AC	DD 000AC	PUSHL	FRAME_WIDTH	6743
		0200	C3	C5 000AF	MULL3	512(R3), LINE_NO, R0	6742
			0C	BC40 9F 000B5	PUSHAB	FRAME_PTR[R0]	
			14	AE 9F 000B9	PUSHAB	STRING_DESC	6741
			04	AC DD 000BC	PUSHL	SCB	6742
BB	FD63	CF	05	FB 000BF	CALLS	#5, CENTER_FRAME	
	08	AE	0200	8F 3C 000C4	MOVZWL	#512, STRING_DESC	6745
			52	F2 000CA	A0BLSS	NUM_LINES, LINE_NO, 5\$	6725
			50	DO 000CE	MOVL	NUM_LINES, R0	6748
			54	D4 000D1	RET		
			50	D4 000D2	CLRL	R0	6749
			04	000D4	RET		

; Routine Size: 213 bytes, Routine Base: CODE + 30D0

6750 1 %sbttl 'FIND_DEST_PTR - Finds an empty Position in the Frame'
6751 1 ++
6752 1 Functional Description:
6753 1 This routine finds the first empty frame position from the bottom
6754 1 of the frame. The returned parameters include the dest_ptr (position
6755 1 found) and the length left in the frame. If unable to find an empty
6756 1 position then RET_OFFSET = .FRAME_LENGTH
6757 1
6758 1 Formal Parameters:
6759 1 SCB
6760 1 CLR STR
6761 1 FRAME PTR
6762 1 FRAME LENGTH
6763 1 FRAME WIDTH
6764 1 RETURN PTR
6765 1 RET_LEN
6766 1
6767 1 Implicit Inputs:
6768 1 none
6769 1
6770 1 Implicit Outputs:
6771 1 none
6772 1
6773 1 Returned Value:
6774 1 none
6775 1
6776 1 Side Effects:
6777 1 Truncation is possible.
6778 1
6779 1 --
6780 1 ROUTINE FIND_DEST_PTR {
6781 1 SCB : REF \$BBLOCK,
6782 1 CLR STR : REF VECTOR[2],
6783 1 FRAME_PTR : REF PAGE_ARRAY,
6784 1 FRAME_WIDTH : Number of Columns
6785 1 FRAME_LENGTH : Number of Rows
6786 1 RET_OFFSET : REF VECTOR
6787 1) : NOVALUE =
6788 2 BEGIN
6789 2 LOCAL
6790 2 CURR_PTR :
6791 2
6792 2 exit if frame length is zero
6793 2
6794 2 IF .FRAME_LENGTH EQL 0
6795 2 THEN
6796 2 RETURN;
6797 2
6798 2 DECR I FROM (.FRAME_LENGTH-1) TO 0
6799 2 DO
6800 2 BEGIN
6801 2 CURR_PTR = FRAME_PTR[0..I..SCB[PSMSL_PAGE_WIDTH]]:
6802 2
6803 2 IF CHSEQC .FRAME_WIDTH, CURR_PTR,
6804 2 i, CHSPTR(UPLIT ('-')), XC' ')
6805 2 THEN
6806 2 BEGIN

SEPARATE
V04-001

Print Symbiont -- separation routines
FIND_DEST_PTR - Finds an empty Position in the

E 12
16-Sep-1984 02:23:03
14-Sep-1984 22:32:26

VAX-11 Bliss-32 V4.0-742
[PRTSMB.SRC]SEPARATE.B32;2

Page 195
(49)

: 5908 6807 4 RET_OFFSET[0] = .I;
: 5909 6808 4 RETURN;
: 5910 6809 3 END;
: 5911 6810 2
: 5912 6811 2
: 5913 6812 2 RET_OFFSET[0] = 0;
: 5914 6813 2 ! exit with 0 if destination not found
: 5915 6814 1 END;

00 00 00 20 031A5 .BLKB 3
00 00 00 20 031A8 P.AGN: .ASCII \ \<0><0><0>

007C 00000 FIND_DEST_PTR:
01 50 54 0200 C5 C5 00011 18: .WORD Save R2,R3,R4,R5,R6
56 50 0C AC C1 00017 18: TSTL FRAME_LENGTH
20 66 10 AC 2D 0001C 18: BEQL 3\$
D8 AF 00022 18: MOVL SCB, R5
05 12 00024 18: MOVL FRAME_LENGTH, I
BC 54 D0 00026 18: BRB 2\$
E3 18 54 F4 0002B 28: MULL3 512(R5), I, R0
BC D4 0002E 18: ADDL3 FRAME_PTR, R0, CURR_PTR
04 00031 38: CMPC5 FRAME_WIDTH, (CURR_PTR), #32, #1, P.AGN
BNEQ 2\$
MOVL I, @RET_OFFSET
RET
SOBGEQ I, 1\$
CLRL @RET_OFFSET
RET

: Routine Size: 50 bytes, Routine Base: CODE + 31AC

```
: 5917    6815 1 %sbttl 'FIND_SOURCE_PTR - Finds an empty Position in the Frame'
: 5918    6816 1 ++
: 5919    6817 1 Functional Description:
: 5920    6818 1 This routine finds the first nonempty frame position from the bottom
: 5921    6819 1 of the frame. The returned parameters include the ret_str descriptor
: 5922    6820 1 return_ptr (position found) and the length left in the frame. If
: 5923    6821 1 unable to find a string position then RET_STR[SIZE]=0, RET_LEN = 0 and
: 5924    6822 1 RETURN_PTR = FRAME_PTR.
: 5925    6823 1
: 5926    6824 1 Formal Parameters:
: 5927    6825 1      SCB
: 5928    6826 1      RET_STR
: 5929    6827 1      FRAME_PTR
: 5930    6828 1      FRAME_LENGTH
: 5931    6829 1      FRAME_WIDTH
: 5932    6830 1      RET_OFFSET
: 5933    6831 1
: 5934    6832 1      Implicit Inputs:
: 5935    6833 1          none
: 5936    6834 1
: 5937    6835 1      Implicit Outputs:
: 5938    6836 1          none
: 5939    6837 1
: 5940    6838 1      Returned Value:
: 5941    6839 1          none
: 5942    6840 1
: 5943    6841 1      Side Effects:
: 5944    6842 1          Truncation is possible.
: 5945    6843 1
: 5946    6844 1      --
: 5947    6845 1      ROUTINE FIND_SOURCE_PTR (
: 5948    6846 1          SCB           : REF $BLOCK,
: 5949    6847 1          RET_STR        : REF VECTOR[2],
: 5950    6848 1          FRAME_PTR     : REF PAGE_ARRAY,
: 5951    6849 1          FRAME_WIDTH   :
: 5952    6850 1          FRAME_LENGTH  :
: 5953    6851 1          RET_OFFSET    : REF VECTOR
: 5954    6852 1          ) : NOVALUE =
: 5955    6853 2      BEGIN
: 5956    6854 2
: 5957    6855 2      LOCAL
: 5958    6856 2          CURR_PTR;
: 5959    6857 2
: 5960    6858 2
: 5961    6859 2      ! exit if frame length is zero
: 5962    6860 2
: 5963    6861 2      IF .FRAME_LENGTH EQ 0
: 5964    6862 2      THEN
: 5965    6863 3          BEGIN
: 5966    6864 3          RET_STR[SIZE] = 0;
: 5967    6865 3          RETURN;
: 5968    6866 2          END;
: 5969    6867 2
: 5970    6868 2      DECR I FROM (.FRAME_LENGTH-1) TO 0
: 5971    6869 2      DO
: 5972    6870 3          BEGIN
: 5973    6871 3          CURR_PTR = FRAME_PTR[0..I..SCB[PSMSL_PAGE_WIDTH]];
```

```

5974 6872 3
5975 6873 3 IF CHSNEQ( .FRAME_WIDTH, CURR_PTR,
5976 6874 3 , CHSPTR( UPLIT ('-')), XC' ')
5977 6875 3 THEN
5978 6876 4 BEGIN
5979 6877 4 CHSMOVE(.FRAME_WIDTH, CURR_PTR,
5980 6878 4 , RET_STR[ADDR]);
5981 6879 4 RET_STR[SIZE] = .FRAME_WIDTH;
5982 6880 4 RET_OFFSET[0] = .I;
5983 6881 4 RETURN; ! exit this routine
5984 6882 3 END;
5985 6883 2
5986 6884 2
5987 6885 2 RET_STR[SIZE] = 0;
5988 6886 2 RET_OFFSET = 0;
5989 6887 2
5990 6888 1 END;

```

00 00 00 20 031DE 031E0 P.AGO: .BLKB 2
.ASCII \ \<0><0><0>

03FC 00000 FIND_SOURCE_PTR:									
									.WORD Save R2,R3,R4,R5,R6,R7,R8,R9
								TSTL FRAME_LENGTH	6845
								BNEQ 1\$	6861
								CLRL @RET_STR	6864
								RET	6863
								MOVL SCB, R8	6871
								MOVL FRAME_LENGTH, I	6873
								BRB 3\$:
								MULL3 512(R8), I, R0	6871
								ADDL3 FRAME_PTR, R0, CURR_PTR	6873
								CMPC5 FRAME_WIDTH, (CURR_PTR), #32, #1, P.AGO	6873
								BEQL 3\$:
								MOVL RET_STR, R7	6878
								MOV3 FRAME_WIDTH, (CURR_PTR), @4(R7)	6879
								MOVL FRAME_WIDTH, (R7)	6880
								MOVL I, @RET_OFFSET	6876
								RET	6868
								SOBGEQ I, 2\$	6885
								CLRL @RET_STR	6886
								CLRL RET_OFFSET	6888

; Routine Size: 71 bytes, Routine Base: CODE + 31E4

5992 6889 1 %sbttl 'DELIMIT_STRING - Return the last position of this delimiter'
5993 6890 1 ++
5994 6891 1 Functional Description:
5995 6892 1 This procedure returns the position of the delimiter nearest the
5996 6893 1 string_end
5997 6894 1 Formal Parameters:
5998 6895 1 STR_PTR
5999 6896 1 CHAR
6000 6897 1 STR_END - Pointer of String to delimit
6001 6898 1 - Character delimiter
6002 6899 1 - End position of string
6003 6900 1 Implicit Inputs:
6004 6901 1 none
6005 6902 1 Implicit Outputs:
6006 6903 1 none
6007 6904 1 Returned Value:
6008 6905 1 none
6009 6906 1 Side Effects:
6010 6907 1 none
6011 6908 1 none
6012 6909 1 --
6013 6910 1 ROUTINE DELIMIT_STRING (
6014 6911 1 STR_PTR,
6015 6912 1 CHAR,
6016 6913 1 STR_END) =
6017 6914 1 BEGIN LOCAL
6018 6915 1 POS,
6019 6916 2 BASE,
6020 6917 2 TEMP_PTR,
6021 6918 2 CHAR_PTR;
6022 6919 2 TEMP_PTR = .STR_PTR;
6023 6920 2 BASE = .STR_PTR + .STR_END - 1;
6024 6921 2 CHAR_PTR = CHSPTR(CHAR);
6025 6922 2 DECR CURR_PTR FROM (.BASE) TO .STR_PTR DO
6026 6923 2 BEGIN
6027 6924 2 TEMP_PTR = CHSPTR(.CURR_PTR);
6028 6925 2 POS = CHSEQL(1, .TEMP_PTR, 1, .CHAR_PTR);
6029 6926 2 IF (.POS EQL 1) AND
6030 6927 2 (.CURR_PTR EQL .STR_PTR)
6031 6928 2 THEN RETURN .STR_END; ! ...return the original length
6032 6929 2 IF (.POS EQL 1) AND
6033 6930 2 (.CURR_PTR GTR .STR_PTR)
6034 6931 2 THEN RETURN (.CURR_PTR - .STR_PTR + 1); ! char in string
6035 6932 2 ! ...return position plus one
6036 6933 2 IF (.POS EQL 0) AND (.CURR_PTR EQL .STR_PTR) THEN
6037 6934 2 RETURN .STR_END; ! handle extra decrement
6038 6935 2 END;

SEPARATE
V04-001

Print_Symbiont -- separation routines
DELIMIT_STRING - Return the last position

I 12
16-Sep-1984 02:23:03
14-Sep-1984 22:32:26

VAX-11 Bliss-32 V4.0-742
[PRTSMB.SRC]SEPARATE.B32;2

Page 199
(51)

6049 6946 2 RETURN .STR_END; ! ...return the original length
6050 6947 2
6051 6948 2
6052 6949 1 END;

003C 00000 DELIMIT_STRING:

50	52	04	AC	D0	00002	.WORD	Save R2,R3,R4,R5	6912	
	54	52	DO	00006		MOVL	STR_PTR R2	6923	
	52	0C	AC	C1	00009	MOVL	R2_TEMP_PTR		
		50	D7	0000E		ADDL3	STR_END,-R2, R0	6924	
	55	08	AC	9E	00010	DECL	BASE		
		3A	11	00014		MOVAB	CHAR, CHAR_PTR	6925	
	54	50	D0	00016	1\$:	BRB	6S	6931	
		51	D4	00019		MOVL	CURR_PTR, TEMP_PTR	6929	
	65	64	91	0001B		CLRL	R1		
		02	12	0001E		(CMPB	(TEMP_PTR), (CHAR_PTR)	6931	
		51	D6	00020		BNEQ	2\$		
	53	51	D0	00022	2\$:	INCL	R1		
		51	D4	00025		MOVL	R1, POS		
	01	53	D1	00027		CLRL	R1	6933	
		07	12	0002A		(CMPL	POS, #1		
		51	D6	0002C		BNEQ	3\$		
	52	50	D1	0002E		INCL	R1		
		22	13	00031		(CMPL	CURR_PTR, R2	6934	
	0F	51	E9	00033	3\$:	BEQL	7\$		
	52	50	D1	00036		BLBC	R1, 4\$	6938	
		0A	15	00039		(CMPL	CURR_PTR, R2	6939	
	50	52	C3	0003B		BLEQ	4\$		
		51	D6	0003F		SUBL3	R2, CURR_PTR, R1	6941	
	50	51	D0	00041		INCL	R1		
			04	00044		MOVL	R1, R0		
			53	D5	00045	4\$:	RET		
			05	12	00047		TSTL	POS	6943
	52	50	D1	00049		BNEQ	5\$		
		07	13	0004C		(CMPL	CURR_PTR, R2		
	52	50	D7	0004E	5\$:	BEQL	7\$	6927	
		50	D1	00050	6\$:	DECL	CURR_PTR		
		C1	18	00053		(CMPL	CURR_PTR, R2		
	50	0C	AC	D0	00055	BGEQ	1\$		
			04	00059	7\$:	MOVL	STR_END, R0	6947	
						RET		6949	

; Routine Size: 90 bytes, Routine Base: CODE + 322B

J 12
Print Symbiont -- separation routines 16-Sep-1984 02:23:03 VAX-11 Bliss-32 V4.0-742
DELIMIT_STRING_NOT - Return the last position o 14-Sep-1984 22:32:26 [PRTSMB.SRC]SEPARATE.B32:2

6054 6950 1 %sbttl 'DELIMIT_STRING_NOT - Return the last position of not this delimiter'
6055 6951 1 ++
6056 6952 1 Functional Description:
6057 6953 1 This procedure returns the length of the string without the delimited
6058 6954 1 characters on the string end. Return the original length if
6059 6955 1 non_delimiters cannot be found.
6060 6956 1
6061 6957 1 Formal Parameters:
6062 6958 1 STR_PTR - Pointer of String to delimit
6063 6959 1 CHAR - Character delimiter
6064 6960 1 STR_END - End position of string
6065 6961 1
6066 6962 1 Implicit Inputs:
6067 6963 1 none
6068 6964 1
6069 6965 1 Implicit Outputs:
6070 6966 1 none
6071 6967 1
6072 6968 1 Returned Value:
6073 6969 1 none
6074 6970 1
6075 6971 1 Side Effects:
6076 6972 1 none
6077 6973 1 ---
6078 6974 1 ROUTINE DELIMIT_STRING_NOT(
6079 6975 1 STR_PTR,
6080 6976 1 CHAR
6081 6977 1 STR_END) =
6082 6978 2 BEGIN
6083 6979 2
6084 6980 2
6085 6981 2 LOCAL
6086 6982 2 PTR : REF VECTOR[,byte];
6087 6983 2
6088 6984 2 IF .STR_END EQL 0
6089 6985 2 THEN
6090 6986 2 RETURN 0;
6091 6987 2
6092 6988 2 PTR = .STR_PTR + .STR_END - 1;
6093 6989 2
6094 6990 2 WHILE .PTR GTRU .STR_PTR
6095 6991 2 DO
6096 6992 2 IF .PTR[0] NEQU .CHAR
6097 6993 2 THEN
6098 6994 2 EXITLOOP
6099 6995 2 ELSE
6100 6996 2 PTR = .PTR - 1;
6101 6997 2
6102 6998 2 RETURN .PTR - .STR_PTR + 1;
6103 6999 2
6104 7000 1 END;

0000 00000 DELIMIT_STRING_NOT:

SEPARATE
V04-001

Print Symbiont -- separation routines
DELIMIT_STRING_NOT - Return the last p

K 12

16-Sep-1984 02:23:03
14-Sep-1984 22:32:26

VAX-11 Bliss-32 V4.0-742
[PRTSMB.SRC]SEPARATE.B32;2

Page 201
(52)

; Routine Size: 39 bytes, Routine Base: CODE + 3285

```
7001 1 %sbttl 'DISCARD - Returns a Pointer to First Char NOT Discarded'
7002 1 ++
7003 1 Functional Description:
7004 1 This routine discards the character in the string from the beginning
7005 1 (LEADING) or end(TRAILING) of the string(STR_PTR) and returns a pointer
7006 1 to the first position that is found not to contain the discard
7007 1 character(HAR). The boundaries of the string are the beginning pointer
7008 1 and the string length(LEN).
7009 1
7010 1 Formal Parameters:
7011 1 WHICH_WAY
7012 1 CHAR
7013 1 STR_PTR
7014 1 LEN
7015 1 RET_PTR
7016 1
7017 1 Implicit Inputs:
7018 1 none
7019 1
7020 1 Implicit Outputs:
7021 1 none
7022 1
7023 1 Returned Value:
7024 1 none
7025 1
7026 1 Side Effects:
7027 1 none
7028 1 --
7029 1 ROUTINE DISCARD (
7030 1 WHICH_WAY,
7031 1 CHAR,
7032 1 STR_PTR ,
7033 1 LEN : WORD, ! force word size
7034 1 RET_LEN : REF VECTOR,
7035 1 RET_PTR : REF VECTOR) : NOVALUE =
7036 2 BEGIN
7037 2
7038 2 LITERAL LEADING = 0;
7039 2
7040 2 LOCAL DSTR_DESC : VECTOR[2];
7041 2
7042 2 ! a descriptor is needed for BASSEDIT routine
7043 2 DSTR_DESC[SIZE] = .LEN;
7044 2 DSTR_DESC[ADDR] = .STR_PTR;
7045 2
7046 2
7047 2 !F .WHICH_WAY EQL LEADING THEN
7048 2 BEGIN
7049 2 BASSEDIT ( DSTR_DESC[0], DSTR_DESC[0], LEAD_MASK); ! trim leading blanks
7050 2 RET_PTR[0] = .DSTR_DESC[ADDR];
7051 2 RET_LEN[0] = .DSTR_DESC[SIZE];
7052 2 END
7053 2
7054 2 ELSE ! trim trailing blanks
7055 2 BEGIN
7056 2 RET_LEN[0] = DELIMIT STRING_NOT ( .STR_PTR, .CHAR, .LEN);
7057 2 RET_PTR[0] = .STR_PTR;
```

SEPARATE
V04-001

Print Symbiont -- separation routines
DISCARD - Returns a Pointer to First Char NOT D

M 12

16-Sep-1984 02:23:03
14-Sep-1984 22:32:26

VAX-11 Bliss-32 V4.0-742
[PRTSMB.SRC]SEPARATE.B32;2

Page 203
(53)

: 6163
: 6164

7058 2 END:
7059 1 END:

				0000 000000 DISCARD: WORD	Save nothing	
		04	5E	04 C2 00002	SUBL2 #4, SP	7029
			7E	10 AC 3C 00005	MOVZWL LEN, DSTR_DESC	7045
		04	AE	0C AC DD 00009	MOVL STR_PTR, DSTR_DESC+4	7046
				04 AC D5 0000E	TSTL WHICH_WAY	7048
				19 12 00011	BNEQ 1\$	
				08 DD 00013	PUSHL #8	7050
				06 AE 9F 00015	PUSHAB DSTR_DESC	
				08 AE 9F 00018	PUSHAB DSTR_DESC	
	00000000G	00		03 FB 0001B	CALLS #3, BASSEDIT	
		18	BC	04 AE DD 00022	MOVL DSTR_DESC+4, RET_PTR	7051
		14	BC	6E DD 00027	MOVL DSTR_DESC, RET_LEN	7052
				04 0002B	RET	7048
		7E		10 AC 3C 0002C	1\$: MOVZWL LEN, -(SP)	7056
				08 AC DD 00030	PUSHL CHAR	
				0C AC DD 00033	PUSHL STR_PTR	
		9F	AF	03 FB 00036	CALLS #3, DELIMIT_STRING_NOT	
		14	BC	50 DO 0003A	MOVL R0, RET_LEN	
		18	BC	0C AC DO 0003E	MOVL STR_PTR, RET_PTR	7057
				04 00043	RET	7059

: Routine Size: 68 bytes, Routine Base: CODE + 32AC

SEPARATE
V04-001

Print Symbiont -- separation routines
FILE_OPEN - Boolean Valued routine indicating f

N 12

16-Sep-1984 02:23:03
14-Sep-1984 22:32:26

VAX-11 Bliss-32 V4.0-742
[PRTSMB.SRC]SEPARATE.B32;2

Page 204
(54)

```
: 6166    7060 1 %sbttl 'FILE_OPEN - Boolean Valued routine indicating file open status'  
: 6167    7061 1 ++  
: 6168    7062 1 Functional Description:  
: 6169    7063 1 This routine interrogates the FAB and determines if the current file  
: 6170    7064 1 is open and/or if information can be extracted from the file.  
: 6171    7065 1 TRUE = 1, FALSE = 0;  
: 6172    7066 1  
: 6173    7067 1 Formal Parameters:  
: 6174    7068 1      none  
: 6175    7069 1  
: 6176    7070 1 Implicit Inputs:  
: 6177    7071 1      none  
: 6178    7072 1  
: 6179    7073 1 Implicit Outputs:  
: 6180    7074 1      none  
: 6181    7075 1  
: 6182    7076 1 Returned Value:  
: 6183    7077 1      none  
: 6184    7078 1  
: 6185    7079 1 Side Effects:  
: 6186    7080 1      none  
: 6187    7081 1 --  
: 6188    7082 1 ROUTINE FILE_OPEN {  
: 6189    7083 1      SCB : REF $BBLOCK ! SCB  
: 6190    7084 1      ) =  
: 6191    7085 2 BEGIN  
: 6192    7086 2  
: 6193    7087 2 RETURN .SCB[PSMSV_FAB_VALID]  
: 6194    7088 2  
: 6195    7089 1 END:
```

0000 00000 FILE_OPEN:
50 10 A0 01 04 AC D0 00002 .WORD Save nothing
 04 EF 00006 MOVL SCB, R0
 04 0000C EXTZV #4, #1, 16(R0), R0
 RET

: Routine Size: 13 bytes, Routine Base: CODE + 32F0

: 7082
: 7087
: 7089

SEPARATE
V04-001

Print Symbiont -- separation routines
FILE_OPEN - Boolean Valued routine indicating f

B 13

16-Sep-1984 02:23:03
14-Sep-1984 22:32:26

VAX-11 Bliss-32 V4.0-742
[PRTSMB.SRC]SEPARATE.B32;2

Page 205
(55)

: 6197 7090 1 END
: 6198 7091 0 ELUDOM

SE
VO

PSECT SUMMARY

Name	Bytes	Attributes
DATA	4 NOVEC, WRT, RD ,NOEXE,NOSHR, LCL, REL, CON,NOPIC,ALIGN(2)	
CODE	13053 NOVEC,NOWRT, RD , EXE,NOSHR, LCL, REL, CON,NOPIC,ALIGN(2)	

Library Statistics

File	----- Symbols -----	Pages Mapped	Processing Time
	Total Loaded Percent		
\$_\$255\$DUA28:[SYSLIB]LIB.L32;1	18619 88 0	1000	00:01.9

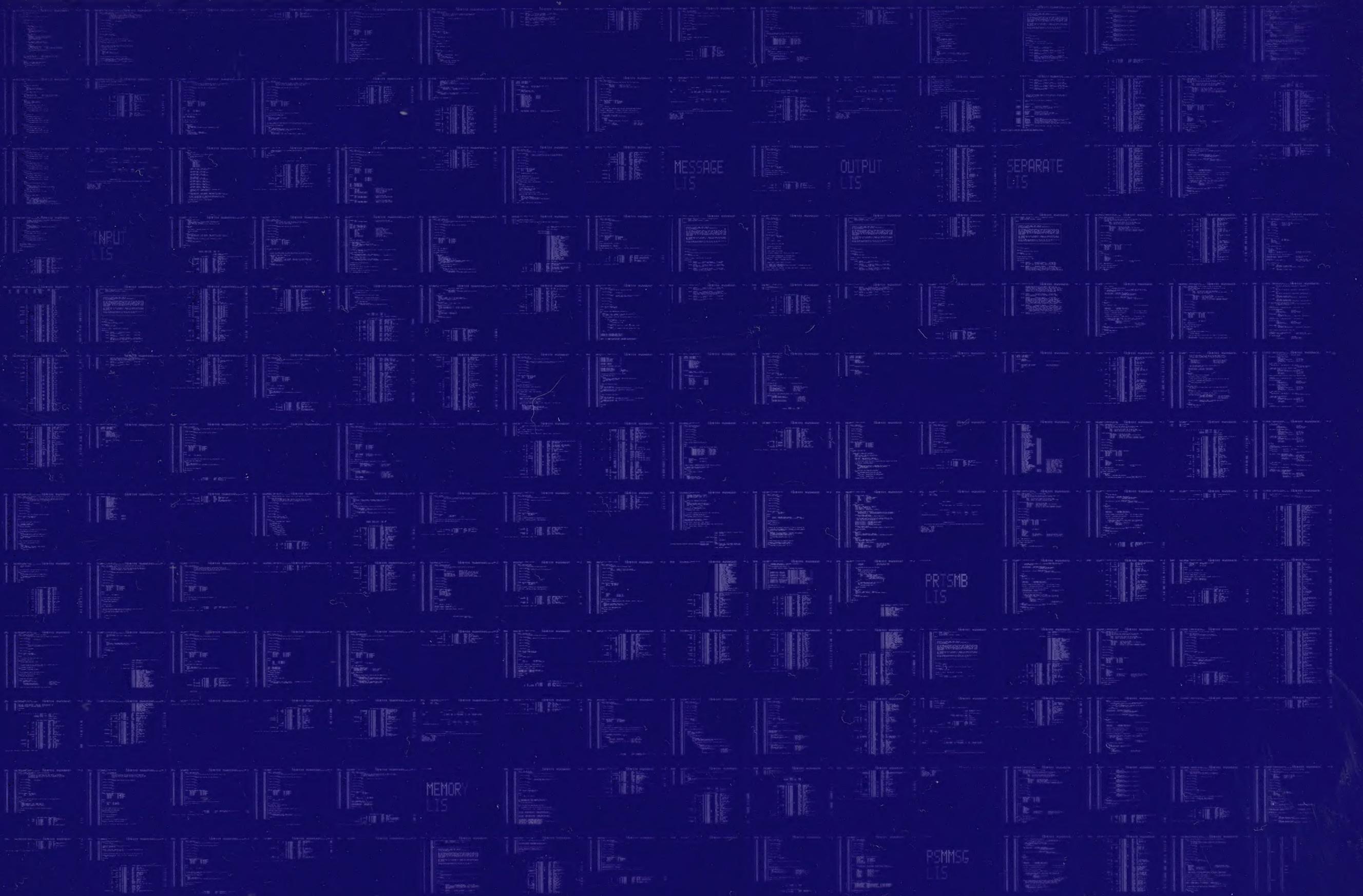
COMMAND QUALIFIERS

BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/LIS=LIS\$:SEPARATE/OBJ=OBJ\$:SEPARATE MSRC\$:SEPARATE/UPDATE=(ENH\$:SEPARATE)

: Size: 10301 code + 2756 data bytes
: Run Time: 03:23.1
: Elapsed Time: 06:52.7
: Lines/CPU Min: 2094
: Lexemes/CPU-Min: 22941
: Memory Used: 682 pages
: Compilation Complete

0310 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY



0311 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

